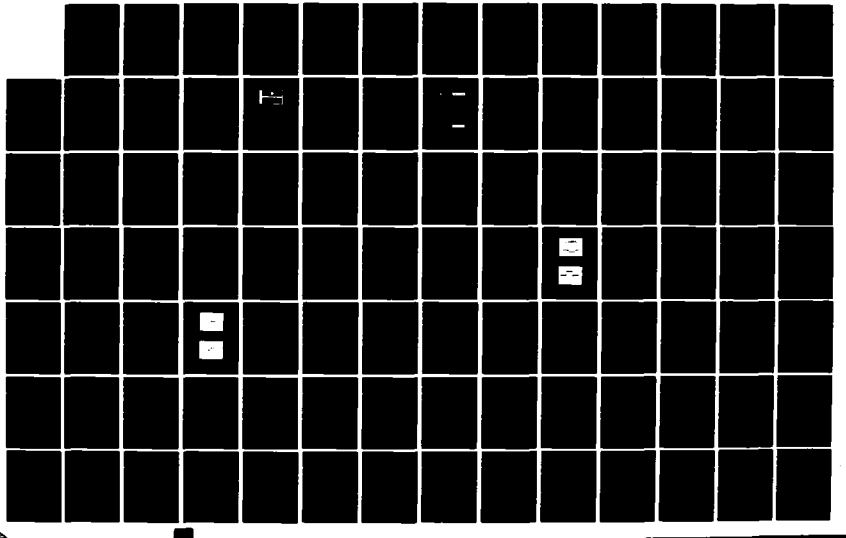
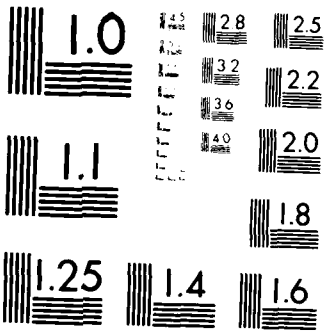
AD-A151 907 DEVELOPMENT OF A DIGITALLY COMPENSATED SAW OSCILLATOR (DCSO) TO PROVIDE T.. (U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. W D COWAN

1/2

UNCLASSIFIED DEC 84 AFIT/GE/ENG/84D-22 F/G 9/5 NL





DEVELOPMENT OF A DIGITALLY COMPENSATED SAW OSCILLATOR (DCSO) TO PROVIDE TEMPERATURE STABLE FREQUENCY

THESIS

AFIT/GE/ENG/84D-22

William D. Cowan
2Lt USAF

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/84D-22

DEVELOPMENT OF A DIGITALLY COMPENSATED SAW OSCILLATOR (DCSO) TO PROVIDE TEMPERATURE STABLE FREQUENCY

THESIS

AFIT/GE/ENG/84D-22

William D. Cowan
2Lt USAF

Approved for public release; distribution unlimited.

AFIT/GE/ENG/84D-22

DEVELOPMENT OF A DIGITALLY COMPENSATED SAW OSCILLATOR (DCSO) TO PROVIDE TEMPERATURE STABLE FREQUENCY

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

William D. Cowan, B.S.
Second Lieutenant, USAF

December 1984

Preface

Though this project was plagued with problems, those same problems made this thesis effort a valuable learning experience in the real world of engineering. On one of the most frustrating days, I was placated by the words, "If everything worked right the first time, we wouldn't need engineers". Those words, spoken by one of the originators of the DCSO, carried me forward many times thereafter.

Throughout the lab work and writing of this thesis, several people have been especially supportive. A special thanks is due my faculty advisor, Roger Colvin, for his guidance and understanding of the nasty microwave problems encountered. Also, a special thanks to Don Smith for his help in obtaining equipment and materials without delay. But most of all, thanks to my wife, Sue, for her outstanding patience, understanding, and ability to sleep with the printer going.

William D. Cowan

Contents

List of Figures

List of Figures

Abstract

A single circuit board Digitally Compensated SAW Oscillator (DCSO) running at 300 MHz demonstrated 30 parts per million temperature stability over a portion of its temperature range. A thick film hybrid controller circuit, for use in the DCSO, was designed and tested. This circuit, referred to as the AFIT IC, consists of two commercial TTL prescalers and a custom NMOS integrated circuit. The TTL prescaler circuits of the AFIT IC were found to be operational, but the failure of the NMOS controller IC to meet design criteria prevented demonstration of a fully integrated DCSO. Off board control circuitry was designed, using portions of the AFIT IC (including part of the NMOS chip), to continue the project. The DCSO was then calibrated and tested over a 0 to 70 degrees Centigrade range. Calibration and testing were facilitated by computer interfaces to the DCSO, an electronic thermometer, and a frequency counter. Despite erroneous temperature behavior of one oscillator loop, compensation to within 30 parts per million over a portion of the temperature range was accomplished for the 300 MHz DCSO.

DEVELOPMENT OF A DIGITALLY COMPENSATED SAW OSCILLATOR (DCSO) TO PROVIDE TEMPERATURE STABLE FREQUENCY

I. Introduction

Background

Surface Acoustic Wave (SAW) oscillators have proven to be a cost effective means of generating precise frequencies. One problem shared by both bulk and SAW oscillators, is that of frequency drift with varying temperature. Several methods of compensating for this temperature dependence have evolved, from ovens designed to keep the crystal temperature constant, to rather elaborate circuits which sense a change in temperature and compensate accordingly. The bulk of these methods require one or more thermistors or thermocouples in contact with the crystal to sense temperature changes. This is not the most effective means of measuring changes in crystal temperature. (7:3-6,16:7)

In 1981, A.J. Slobodnik et. al. demonstrated a Digitally Compensated SAW Oscillator (DCSO) using a novel SAW device to sense changes in crystal temperature. This SAW device incorporates two criss-crossing delay paths on the same substrate. One path, referred to as the clock

path, is aligned with a crystal orientation having low temperature sensitivity. The other path, aligned along a crystal orientation having a high temperature coefficient of delay, is referred to as the thermometer path because its frequency of oscillation provides a direct measure of crystal temperature. A simplified block diagram of the overall DCSO system is shown in Figure I-1. (16)

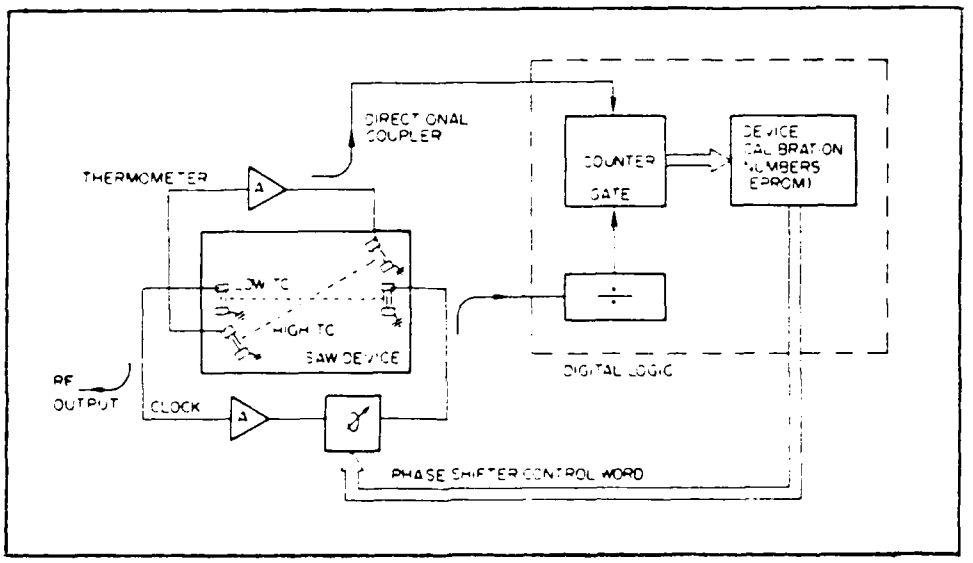


Figure I-1. DCSO Simplified Block Diagram (16:9)

Capt Jerry McGuire in his AFIT thesis work implemented the DCSO circuit for a 300 MHz SAW device on a single circuit board (7). The schematic for this circuit is shown in Figure I-2. A brief description of the circuit and its operation follows.

Both SAW oscillator loops contain one Avantek UTO-524 microwave amplifier (MICamp), one Anzac two way power divider (DS109) used to provide the SAW frequency to a

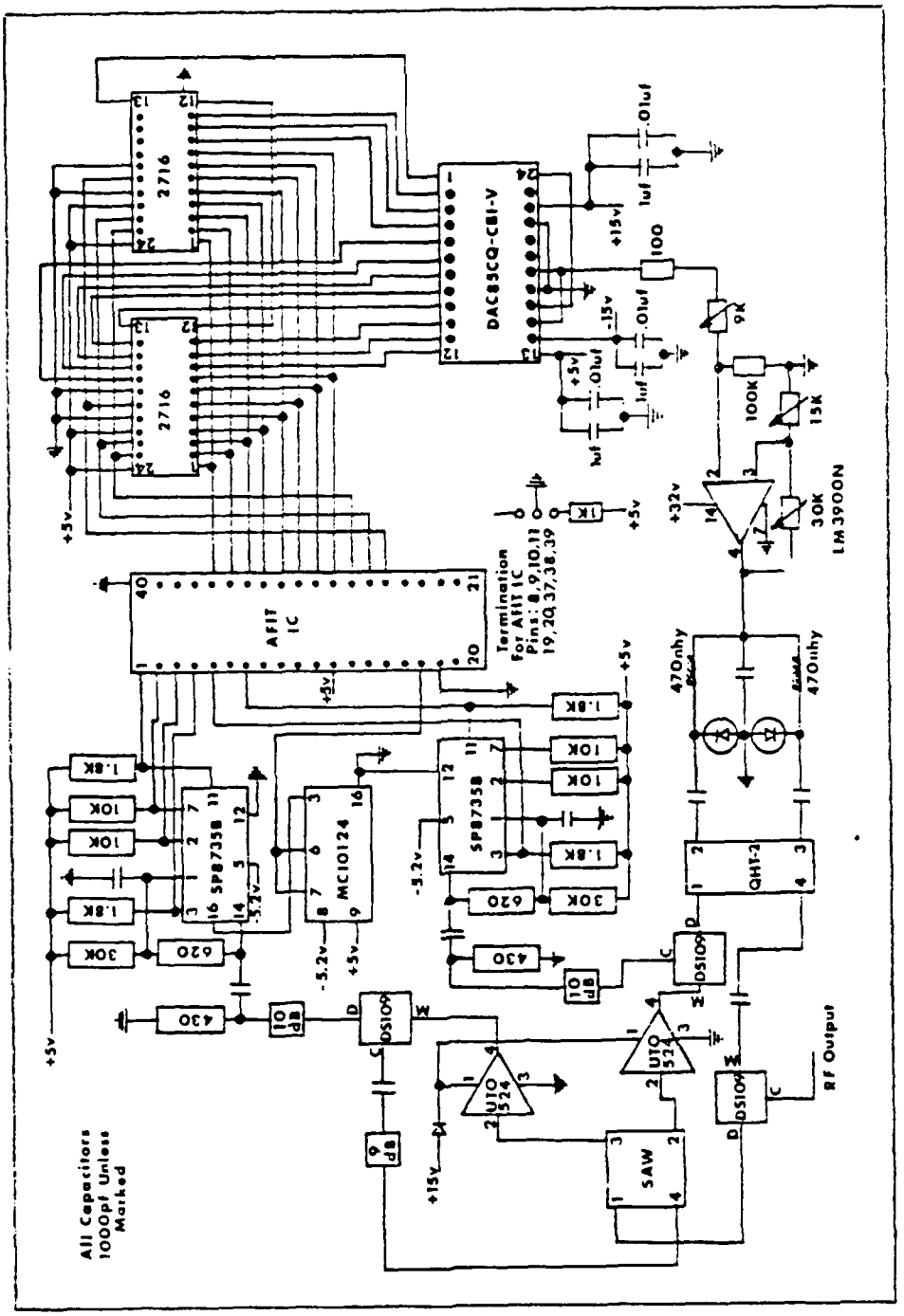


Figure I-2. Single Circuit Board DCSO (7: Sec II, 2)

Plessey SP8735B high speed prescaler, and a SAW delay path. The clock loop contains an additional power divider to provide RF output, and phase shifter circuitry. The phase shifter network consists of two MSI-HA171 varactor diodes, two radio frequency chokes, and three microwave capacitors, coupled to the clock loop through a Merrimac QHT-2 90 degree quadrature hybrid coupler. The DC input voltage to the phase shifter is provided by a LM3900 operational amplifier circuit with a gain of three, which is driven by the Burr Brown DAC85CQ-CBI-V digital-to-analog converter (DAC). The 12 bit digital input for the DAC comes from two parallel 2716 EPROMs whose addresses are provided by the AFIT integrated circuit.

The AFIT IC is best described by looking more closely at DCSO circuit operation (see Figure I-2). The frequency of the clock loop which is held constant over the operating temperature range is prescaled by a Plessey chip and divided by the AFIT IC to provide a gating signal. This gating signal, after conversion to ECL logic levels by the Motorola MC10124 TTL to MECL translator, is applied to the Plessey prescaler of the thermometer loop. This prescaler feeds a counter chain in the AFIT IC. Thus, a counter with latched outputs in the AFIT IC, when gated and reset after each count, serves as a precise measure of temperature. This binary count serves as an address for the EPROMs. In each address location resides a phase shifter control word which corrects the frequency for the temperature induced drift in

the clock loop. Due to the intentional differences in the temperature coefficient of delay of each loop, system convergence is assured.

Summary of Current Knowledge

The RF and phase shifter portions of the AFIT DCSO circuit have been tested. The resonant frequencies of the thermometer and clock loops were determined to be 309 MHz and 298 MHz, respectively. The phase shifter circuit was found to provide 107 degrees of phase shift, allowing the clock oscillator frequency to be varied 210 KHz. Also previously tested was a revised version of the CMOS/SOS gate array AFIT IC designed by a team of AFIT students in 1982 (15). This chip was found to be lacking in several respects (7: Sec IV, 2). The other digital circuitry of the DCSO including the Plessey SP8735B prescalers, and Motorola MC10124 translator remain untested. Finally, the AFIT DCSO has not been tested as a system due to the lack of functional control logic.

Objective

Although major portions of the AFIT DCSO have been designed and tested, the lack of working digital control circuitry, the AFIT IC, has prevented demonstration of a working system. Therefore, a new AFIT IC will be designed, fabricated, tested, and inserted into the DCSO. Once all the other circuits are verified to be operating properly, the system will be calibrated. Finally, its ability to

compensate for drift in frequency will be evaluated as temperature is varied.

Approach

A new version of the AFIT IC will be designed and tested. Due to the availability of CAD tools and fabrication facilities, NMOS technology was chosen for the controller chip. To allow for the relatively slow speed of NMOS, this version of the AFIT IC will be a thick-film hybrid circuit containing two commercial TTL prescalers, and the NMOS controller circuit. This circuit must be designed to work in the existing DCSO circuit board.

Once working control circuitry is obtained, operation of the rest of the DCSO will be verified, and finally the AFIT DCSO will be calibrated and tested. Testing will include both room temperature operation, and temperature cycling in an environmental chamber.

Sequence of Presentation

In Chapter II of this report the design of the AFIT IC thick film hybrid circuit is presented, followed by the design of the NMOS controller chip in Chapter III. Next, the computer system and interfaces used for calibration and testing are described in Chapter IV. Chapter V details testing of the DCSO's major subsystems, while Chapter VI covers calibration and testing of the DCSO as a whole. Conclusions and recommendations for future work follow in Chapter VII.

II. Thick Film Hybrid Design

Prescaler Circuit

The Plessey prescalers of the DCSO circuit divide the oscillator frequencies (309 and 298 MHz) by eight, providing TTL level signals of approximately 37 MHz to the AFIT IC. Because the student designed NMOS controller chip cannot operate at this speed, additional prescaling of the oscillator signals is required. In order to do this without changing the DCSO circuit board, it was decided to make the AFIT IC a thick film hybrid circuit. This circuit consists of two commercial TTL prescalers in leadless chip carrier (LCC) packages and the NMOS DCSO controller circuit as shown in Figure II-1. The assembled thick film hybrid yields a standard sized 40 pin IC package which will continue to be referred to as the AFIT IC.

The TTL integrated circuits chosen for the AFIT IC prescalers are Texas Instruments part number SNJ54S196FK, Decade/bi-quinary presettable counter/latch. This chip was chosen primarily for its speed (100MHz max) and availability in 20 pin LCC package. Each prescaler is connected in a divide-by-ten configuration as shown schematically in Figure II-2. (17:331-337;18)

The SAW oscillator signals are thus divided by a factor of 80 before reaching the NMOS controller chip. It must therefore operate at about 4 MHz. The results of previous

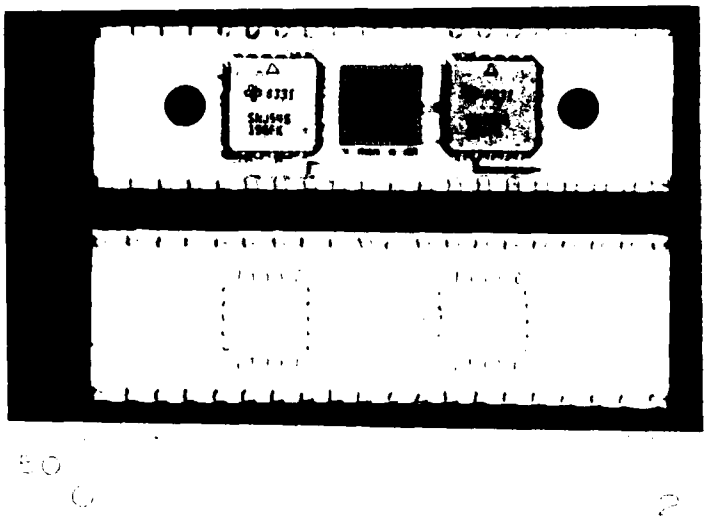


Figure II-1. Thick Film Hybrid AFIT IC

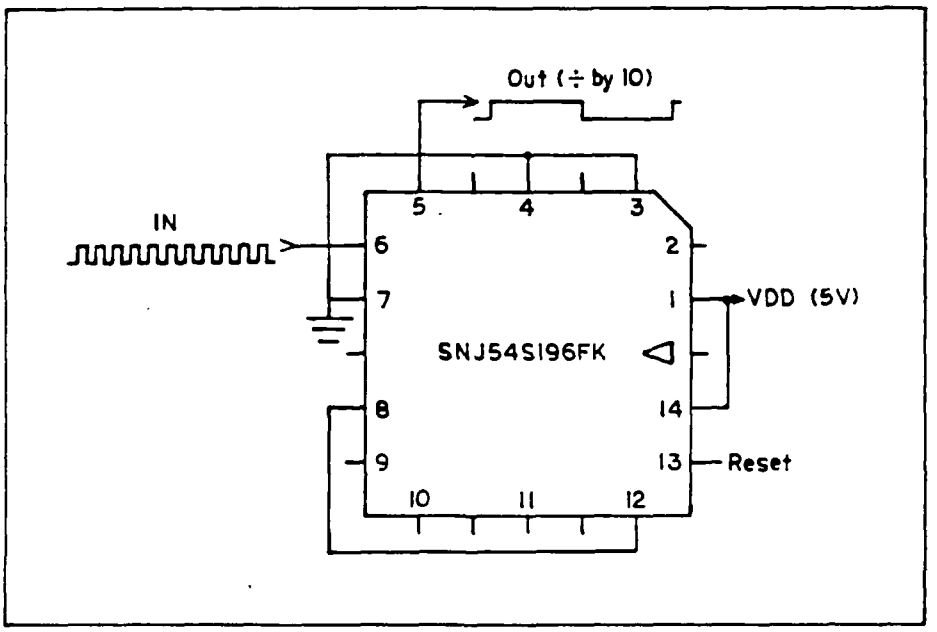


Figure II-2. Thick Film TTL Prescaler Circuit

student chip design efforts indicate this to be quite possible even though the NMOS process is rated at 2 MHz (5: Sec 5 ,7). Due to the prescaling of the oscillator signals, the smallest available deviation in output frequency is 80 Hz. Since 80 Hz is about 0.3 parts per million of the SAW frequency, satisfactory system performance is still expected.

Thick Film Design Constraints

Design of a suitable thick film circuit for the AFIT IC presented several problems. The inputs to the TTL prescaler circuits are approximately 37 MHz signals, requiring microstrip lines for low loss. The size of the substrate was constrained by the current DCSO printed circuit board, as was placement of all inputs and outputs. Solder pads were required for the prescalers and Berg clips, and wire bonding pads were needed for the NMOS controller chip. Because only right angle thick film printing screens were available, all conductor runs were required to be perpendicular. The minimum allowable line width and spacing between lines was 10 mils. Because of the large number of conductor runs required and the size of the two LCCs and NMOS chip, two layers of conductors seperated by a layer of dielectric were used. Each thick film circuit was printed on 0.025 inch thick alumina (relative dielectric = 9.6) substrate two inches square. After visual inspection, good circuits were trimmed to size by laser.

Layout

After designing the thick film circuit, 10 to 1 scale layouts of each layer were accomplished using red, blue, and black tape on mylar film. As layout and taping of the artwork was fairly straightforward, only a brief description of that process is provided before looking at each layer of the thick film circuit in some detail. First, the solder pads for the LCC packages and pin (Berg clip) connections were layed out in red tape. Dimensions for the LCC pads were obtained from a TTL pocket guide (18). On the same sheet of film, bonding pads for the NMOS controller chip were layed out using blue tape. Next, a second sheet of mylar was placed over the first and the first and second conductor layers taped out using red and blue tape respectively. For the most part, one-eighth inch tape was used, yeilding 12.5 mil lines. The width of the microstrip lines was calculated to be 0.0248 inch as shown in Appendix A. Thus, 0.25 inch tape was used for the microstrip lines. Note that neither of the 50 ohm lines is true microstrip because of the dielectric covering the conductor and, in one case, conductors on the second layer crossing over. Due to space limitations, this non-optimum layout was unavoidable. Finally, a third sheet of mylar was placed over the conductor artwork and the dielectric layer taped out using red tape, leaving feedthrough holes where required. The ground plane for the underside of the thick film circuit was

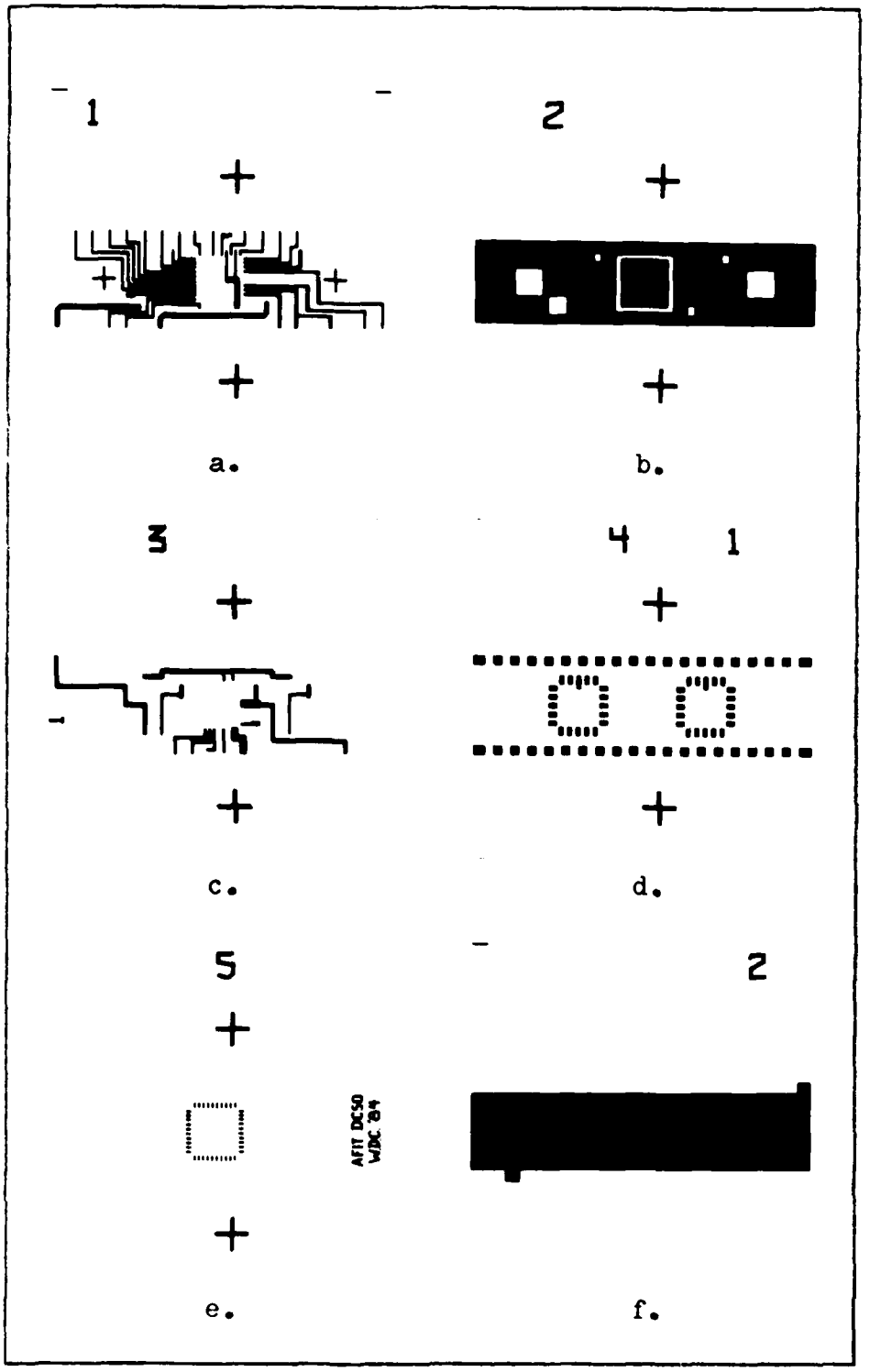


Figure II-3. AFIT IC Thick Film Masks

taped out using blue tape. For the large areas of the dielectric and ground plane layers, only an outline of the regions was taped. These areas were later filled in on the scaled negatives of the artwork using iron oxide. Through photoreduction and color separation, masks for each of the six layers were produced as shown in Figure II-3. Each of these layers will now be discussed in the order in which they were printed.

Thick Film Construction

Materials. The materials used for the AFIT IC hybrid thick film were chosen primarily for their availability, though other characteristics were considered. Specifically, the thermal coefficient of expansion for each material could not differ too greatly from that of the alumina hybrid substrate due to the temperature cycling requirements of the DCSO circuit. A 78% gold, 21% palladium (1% filler) alloy was chosen for the conductors, ground plane, and solder pads of the circuit because of its solderability. An earlier test version of the thick film circuit used gold for the conductor runs with gold-palladium solder pads. However, when soldering to this version, solder often leached the gold from the conductor runs destroying continuity. To provide wire bonding pads for the NMOS controller chip 99% gold (1% filler) was used. The dielectric used to seperate the first and second conductor layers is MFBC-204A produced by Electro Science Laboratories.

Topside Layers. The first layer printed on the thick film substrate is the first conductor layer (Figure II-3a). Gold-palladium was used for this layer. Carefull alignment of the first conductor layer with the substrate edges was necessary for proper registration with the bottom layers later. Most of the conductor runs are on this layer to provide room on the second layer for component mounting. After baking the first conductor layer, the dielectric layer (Figure II-3b) was printed on using a two-way printing. The two-way printing, which uses both a forward and backward stroke of the squeege, promotes a smooth, pinhole free dielectric layer because of the extra ink applied. After the dielectric printing, each hybrid was inspected to insure that the insulator ink did not fill the small feedthrough holes. An X-acto knife was used to carefully remove any dielectric from the feedthrough holes before firing. As with all of the topside layers following the first conductor pattern, the two large alignment marks were used for registration. After firing, the dielectric layer provides a smooth platform approximately 1 mil thick. On top of the dielectric is printed the second gold-palladium conductor layer (Figure II-3c). Wider conductor runs were used in several places on this layer, primarily for the 5 volt supply and ground lines. A gold-palladium solder pad (Figure II-3d) printing follows the second conductor layer. Next, wire bonding pads (Figure II-3e) for the controller chip are printed using gold for its bondability. The final

printing for the topside of the thick film is a solder ink. This layer is printed using the solder pad mask (Figure II-3d) and pretins all soldering areas.

Bottom Layers. To produce a ground plane for the psuedo-microstrip lines, the bottom of the substrate must also be printed with a conductor layer (Figure II-3f). Tabs are provided in the positions of the ground pins of the AFIT IC. By soldering to the bottom of the Berg clips the ground plane is electrically connected to ground. The solder pad mask (Figure II-3d) is also used on the underside to improve the mechanical strength of the Berg clip connections. Gold-palladium is used for both underside printings for its solderability. Tinning of the bottom solder pads must be accomplished manually because the hybrid substrate must be laid flat for oven firing. The edges of the substrate were used to align the printings on the bottom of the substrate with those on top.

Attaching Integrated Cicuits

Due to the limited space on the thick film circuit, construction of a working device must follow a certain order. First, the NMOS controller chip must be cemented to the die bonding area. If the substrate of the controller chip requires grounding (as was found to be the case), the die attach must be conductive and contact the ground bonding pads. After wirebonding the controller chip pads to the appropriate thick film pads the LCC prescalers can be

soldered in place. This is best accomplished by heating the thick film on a hot plate until solder flows and pressing the prescalers into place. Care must be taken to avoid overheating of the thick film during this process because doing so will cause the conductor runs to lift. The Berg clips may be soldered to the substrate using a low wattage soldering iron and extra solder as necessary. A detailed procedure for mounting of thick film components is provided in Appendix D.

III. DCSO Controller Chip Design

Requirements

The digital control circuitry for the AFIT DCSO is implemented on a single NMOS integrated circuit, referred to as the controller chip. Space limitations on the AFIT IC thick film circuit require that the controller die be less than 0.3 inches square. Furthermore, the thick film layout defines the controller chip's pad layout and floor plan. The controller must operate at the prescaled oscillator frequencies of somewhat less than 4 MHz. Flexibility and testability are necessary for ease of interfacing to external circuitry, system calibration, and to provide data for future controller chip designs. The operating temperature range of the controller IC must cover the entire operating range of the DCSO. Packaged ICs are desired for testing, but unpackaged dies are required for mounting on the AFIT IC thick film substrate.

Signals required for the calibration and operation of the DCSO circuit are described in Table III-1. In comparing this table with McGuire's DCSO circuit diagram (Figure I-2) note that two output signals from the thermometer loop Plessey prescaler (AFIT IC pins 2,3) are not present. Due to the additional TTL prescaling (divide-by-ten) on the hybrid AFIT IC, these signals can no longer be used as valid low order bits of the binary counting chain. All inputs and

outputs to the AFIT IC and controller chip are standard TTL level signals.

Table III-1. Required DCSO Controller Chip Signals

AFIT IC Pin #	Name	Function
1	T-osc	T-Chain counter clock
7	C-osc	C-Chain divider clock
4	T-Reset	Active high reset for thermometer loop Plessey prescaler
N/A	NotT-Reset	Active low reset for thermometer loop TTL prescaler
5	C-Reset	Active high reset for clock loop Plessey prescaler
N/A	NotC-Reset	Active low reset for clock loop TTL prescaler
17	NotGate	Clock inhibit for T-Chain Plessey prescaler via MECL translator
26-36	A0-A10	EPROM Address bits
N/A	Dataready or NotDataready	Either, may be needed for calibration and testing

CAD Tools and Fabrication Facilities

The DCSO controller chip was designed on the AFIT Vax 11/780 computer system. The chip was laid out using CLL (Chip Layout Language) and the Stanford NMOS Cell Library following Mead-Conway design rules with lambda=2.0 microns. The computer generated files describing the integrated circuit were transmitted to Information Sciences Institute of the University of Southern California for manufacturing under the DARPA fabrication service. Requests were made for both packaged and unpackaged chips to be returned.

System Description

Figure III-1 is a functional block diagram of the DCSO controller chip. In conjunction with the timing diagram in Figure III-2, the block diagram fully describes the operational requirements of the controller circuitry. The T-Osc signal, derived from the SAW path with the high temperature coefficient of delay, is used to clock the T-Chain Counter. As the T-Chain counts, the C-Chain Divider which is clocked by C-Osc is also counting. Since long sampling times (on the order of 1 second) are needed to provide more varied vectoring into the EPROM address space (due to large changes in T-Chain count), the C-Chain is much longer than the T-Chain. The control logic used is a direct implementation of the combinatorial logic developed by A.J. Slobodnik for the original TTL version of the DSCO control circuitry (16:31). This circuitry runs off three consecutive bits of the C-Chain and the most significant bit C-Chain bit which also gates the T-Chain.

When the most significant bit of the C-Chain (Gate) goes high, NotGate (low) is applied to an external prescaler on the DCSO board (via the MECL translator which inverts it) stopping the T-chain. Note, that the inversion of the MECL translator was overlooked during design of the chip, and therefore the need for the NotGate signal went unnoticed and it was not implemented. In the ensuing discussion, the correct but nonexistant Notgate will be referred to so future controller chip design efforts will not be confused.

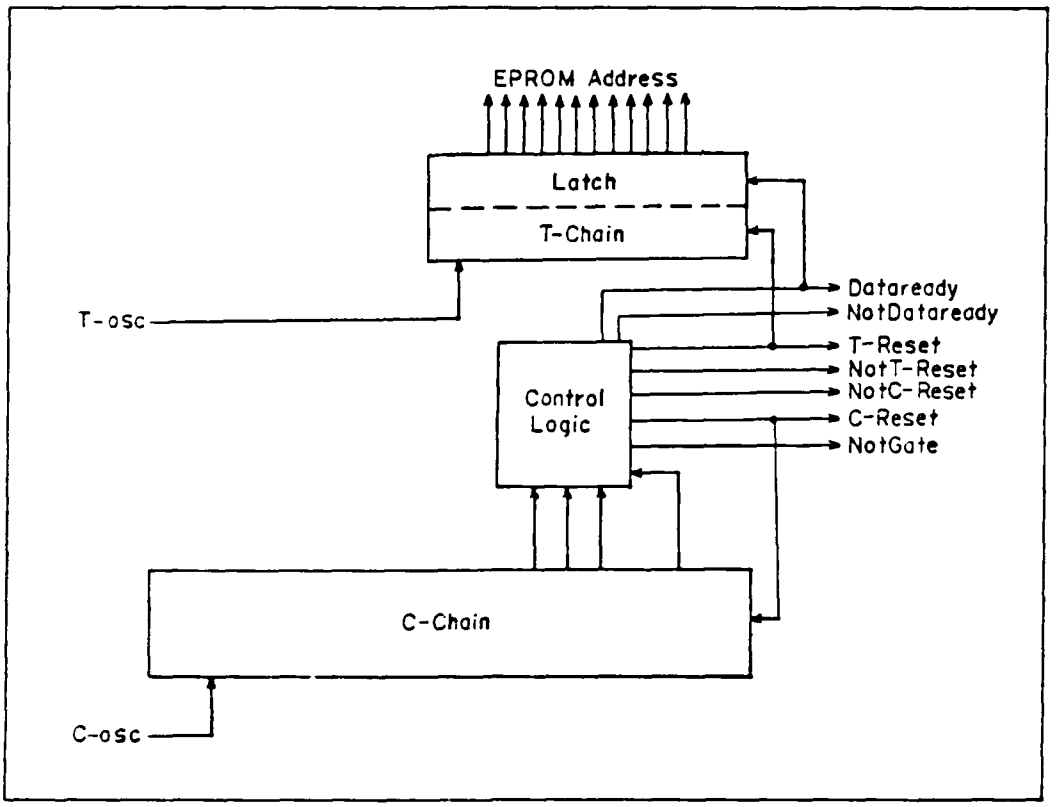


Figure III-1. Block Diagram of DCSO Controller Chip

The Gate signal enables the control logic. Then the latch is strobed by a high on the Dataready line. This strobe allows the T-Chain count to settle in the latch. Next, the T-Chain counter is reset, clearing the count to zero. Finally, the C-Reset goes high setting the C-Chain divider to all zeros, including the most significant bit Gate, thus disabling the control logic until the C-Chain counts up to the point where Gate is high again. At the same time the internal resets occur, the corresponding external prescalers are also reset. A more detailed description of the timing of the control logic can be found in the detailed design section of this report which follows.

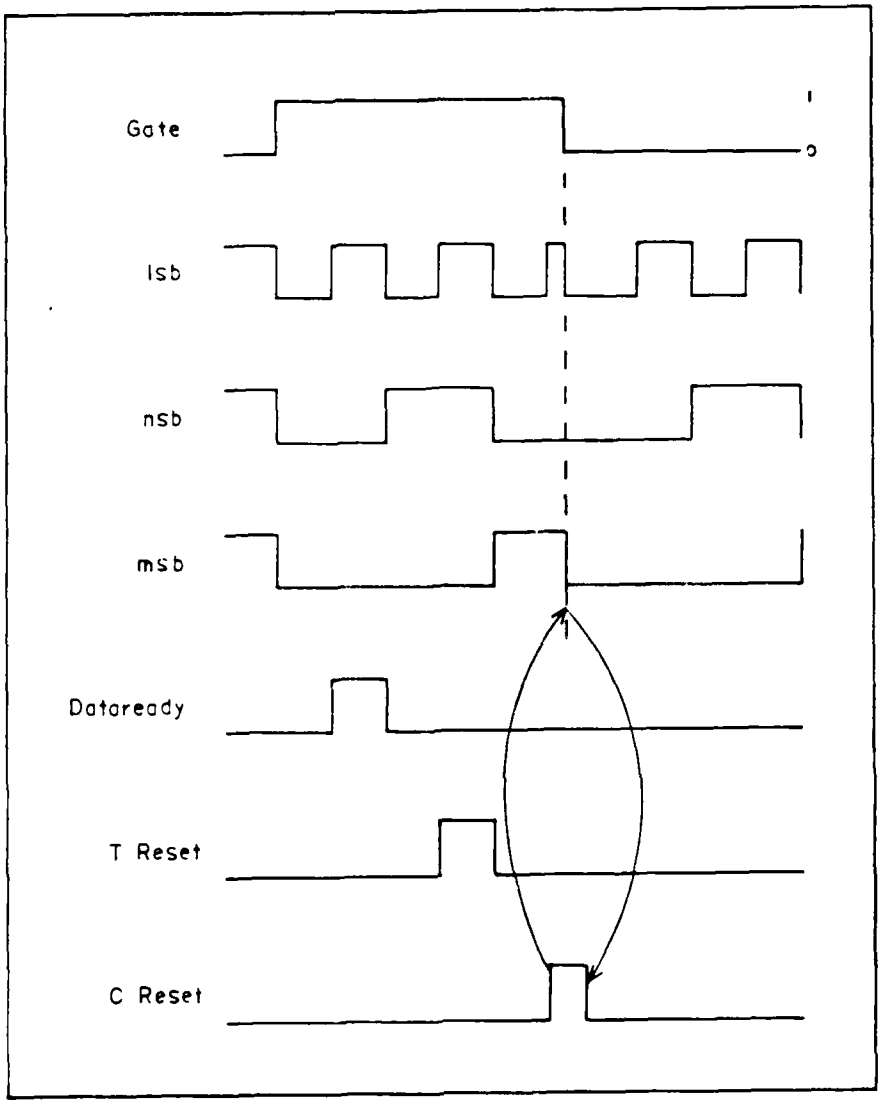


Figure III-2. Controller Chip Timing Diagram

Detailed Design

The major subsystems of the DCSO controller chip are the T-Chain Counter, C-Chain Divider, Data Latch, and the Control Logic. Both the T and C chains (also called Counter and Divider based on how they are used, and for naming

consistent with ealier reports) are made up of the standard library cells Cnt and CntRestore. Each bit of the data latch is made up of three Inverting SB8 standard library cells. In the following sections each subsystem is examined in detail, followed by an overall look at the DCSO controller chip. Figure III-3 shows the logic and pinout of the controller chip (dcso_ver1.00) as well as the approximate floor plan. Further details of controller chip design are provided in Appendix B.

T-Chain Counter and C-Chain Divider. The T-Chain Counter and C-Chain Divider are both ripple-carry counters implemented using the standard library cells Cnt and CntRestore. A CntRestore Cell is used every fourth bit as recommended in the library cell manual (14:71). Each chain is clocked by the two phase clock corresponding to the appropriate oscillator input. To provide some degree of flexibility, the T-Chain is 12 bits long instead of the 11 required to address a 2K byte EPROM (to allow use of a 4K EPROM). Also, the four most significant bits of the C-Chain are brought to output pads. Thus, the length of the C-Chain is adjustable using external jumpers to connect one of these to the Gate input (Gatein). The Gate signal turns off the external prescaler of the T-Chain stopping the T-Chain count and is also an input to the Control Logic. As will be shown later the C-Chain also provides three consecutive bits of its count to the Control Logic for use in generating the control signals.

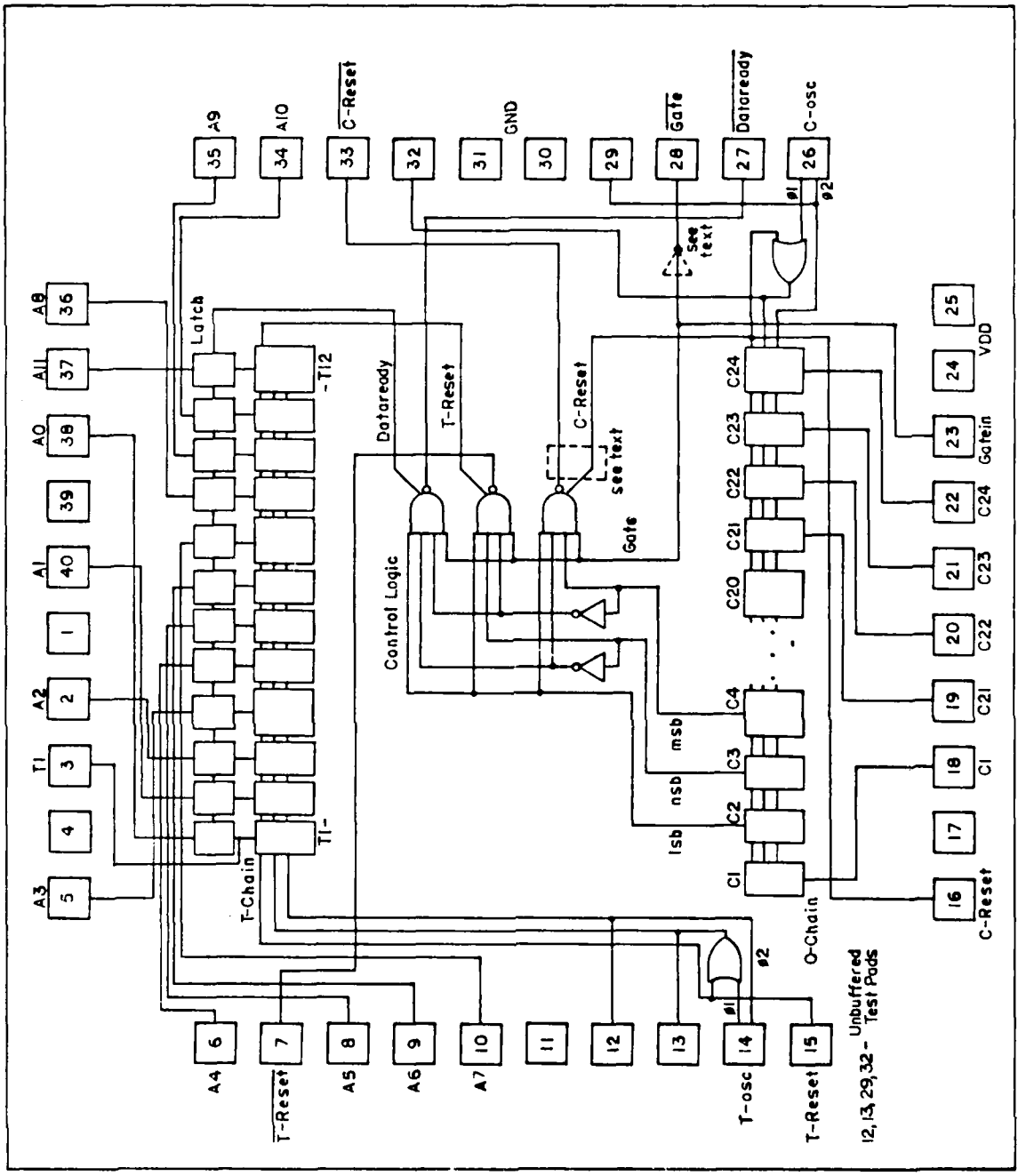


Figure III-3. DCSO Controller Chip Schematic

Data Latch. The circuit diagram for one bit of the 12 bit Data Latch is shown in Figure III-4 (8:224). This latch was successfully implemented using InvertingSB8 standard library superbuffer cells by Hahn/Elam at AFIT in 1983. Twelve of the latch cells generated by Hahn/Elam are used with a common latch enable - Dataready. (4,5)

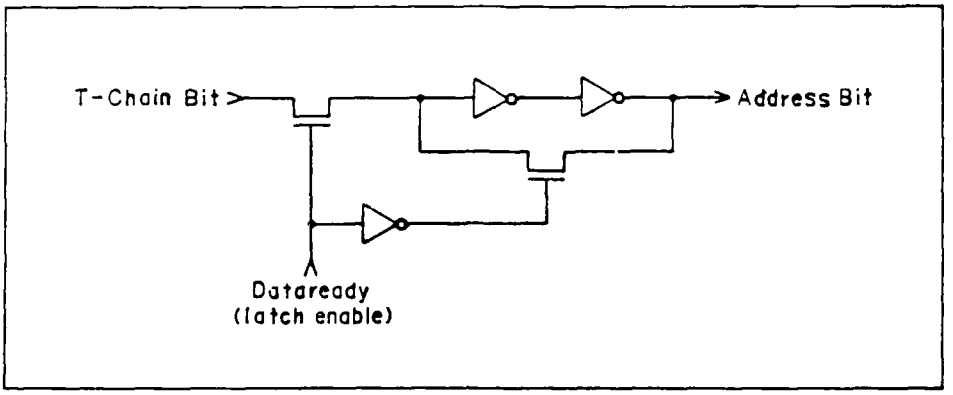


Figure III-4. Latch Cell Circuit Diagram

Control Logic. Because the Control Logic required by the DCSO controller chip is minimal, and no continuous clock is available for a PLA, it is implemented in combinatorial logic. The logic used is the same as that used for the original DCSO control circuit (16:31). The inputs to the logic all come from the C-Chain as shown in Figure III-3. Gate, which is the most significant bit of the C-Chain is an input to all three NAND gates which provide the control logic. Thus, the Gate signal is required before any of the control signals is produced. Other inputs to the NAND gates are LSB, NSB, and MSB or the complements of these signals

which are the outputs from three consecutive stages of the C-Chain. The reader is now referred to the truth table below, the timing diagram, Figure III-2, and the control logic diagrammed in Figure III-3. Recall that when Gate goes high all other stages of the C-Chain have just gone low. As the 3 bits LSB, NSB, MSB count from 000 the control signals are generated as follows:

Signal	LSB	NSB	MSB
Dataready	1	0	0
T-Reset	1	1	0
C-Reset	1	0	1

Note that Dataready and T-Reset are active for only the time that LSB is high. LSB is C-osc divided by four (two divide-by-two counter stages) as shown in Figure III-3. Assumming C-osc = 4 MHz yields:

LSB = C-osc/4 (from circuit)

thus, LSB = 4 Mhz/4 = 1 MHz

since, Period = 1/Frequency

LSB high = Period/2 = 1/(2 X 1 MHz) = 500 nS

Since, the external Plessey prescalers require a reset pulse of at least 100 nS for proper operation, a large margin for error is provided (11:758). Although the logic could be made to run more slowly, extensive "dead time" may degrade overall DCSO performance, because the T-Chain does not count while the control signals are being generated.

DCSO Controller Version 1.00. Figure III-3 shows the entire DCSO Controller Version 1.00 circuit as implemented. A photomicrograph of the fabricated chip is provided in Chapter V. In looking at Figure III-3, one would do well to notice the OR gates in the phase 1 clock lines of both counting chains. Each OR gate has the respective reset line as the second input. The counter chains are reset by grounding the outputs of each counter stage and having one phase 1 pulse. Since the clocks to both chains will be stopped when the reset pulse is generated, it is necessary to insure that phase 1 is high. The OR gates insure that such a condition occurs during resetting, by forcing the phase 1 clock line high if the reset is high. When the reset signal is low, the output of the OR gate simply follows the phase 1 clock signal. The dotted inverter shown in Figure III-3 was not implemented but is required to generate NotGate.

DCSO Controller Version 1.10. The duration of the C-Reset signal is determined solely by hardware delay, because after being generated by the control logic, it changes the inputs to the control logic, turning itself off. This hardware dependence is shown by the two arrows on the timing diagram (Figure III-2). Thus, difficulty in maintaining the external C-Reset signals for the minimum reset time was anticipated. For this reason a second version of the controller chip was produced. Version 1.10 is essentially the same as version 1.00 with the addition of

a one-shot on the output of the NAND gate which generates C-Reset. Figure III-5 is a logic diagram of that one-shot circuit. The position of the one-shot is shown by the dotted rectangle on Figure III-3.

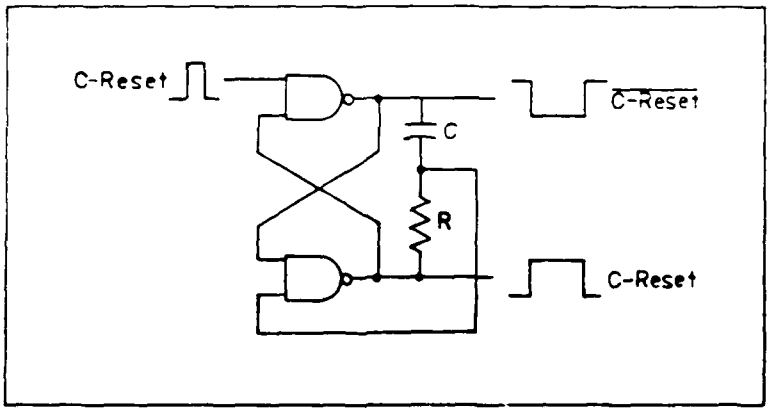


Figure III-5. C-Reset One-Shot Circuit

Floorplan and Pinout. Figure III-3 shows the approximate floor plan for the DCSO controller chip as well as the pinout for version 1.00. Further detail is provided by the photomicrograph in Chapter V. Pinouts for the bonded chips of both versions are provided in Appendix B. For ease of wire-bonding and hence speedier return of fabricated circuits, the DCSO controller is implemented on a MOS Implementation Service (MOSIS) standard 40 pin pad frame. Although this chip is much larger than required by the circuitry, the wasted real estate was traded off for speedy delivery because automated wire bonding is used by the fabrication service for chips with standard pad frames.

DRC/ESIM Results. Prior to submitting any of the chips for fabrication, Design Rule Checks (DRCs) were performed to insure that none of the Mead and Conway design rules had been violated. All versions of the controller chip passed clldrc, alldrc, and fastdrc with no errors. The design was also functionally tested using ESIM, an event driven switch level simulator. The varied simulations showed that all portions of the circuit worked as designed. ESIM failed to show the C-Reset pulse occuring. This is because this simulator does not produce any output until the circuit has stabilized. The occurence of the C-Reset signal was observed indirectly via the outputs of the C-Chain, which were seen to go low when the reset was applied. Results of DRCs and ESIMs are provided in Appendix B.

IV. Automated Test Equipment

Calibration and testing of the DCSO require a microcomputer with appropriate interfacing to a frequency counter and electronic thermometer. In addition, the microcomputer proved to be an invaluable tool in testing specific subsystems of the DCSO circuit. The following sections describe the microcomputer used, frequency counter interface, and electronic thermometer interface.

Computer System

Hardware. The computer provided for work with the AFIT DCSO is an Integrand S-100 bus system with two eight inch disk drives, Heathkit H29 terminal, and Heathkit H25 high speed printer. An Advanced Digital Corporation Super Quad single board computer provides the system's processing power. The Super Quad is a 4 MHz Z80 based microcomputer with two serial ports, two 8 bit parallel ports (Z80 PIO), and 64K of dynamic RAM. Also included on the Super Quad board are a floppy disk controller and Z80 CTC (Clock Timer Circuit). (1)

In order to provide more parallel ports for testing, a Morrow Designs "Switchboard" was added. The Switchboard provides many added features but of primary importance are four switch selectable 8 bit parallel ports. By means of dip switches, the Switchboard allows placement of the parallel ports at various addresses, and designation of each

port as input or output. A base address of 240 decimal was chosen to avoid conflicts with the Super Quad I/O and set via dip switch SW5. Dip switch SW4 sets each port as input (switch off) or output (switch on). At various points in testing, the settings of SW4 were changed as required to provide the parallel ports needed. (8)

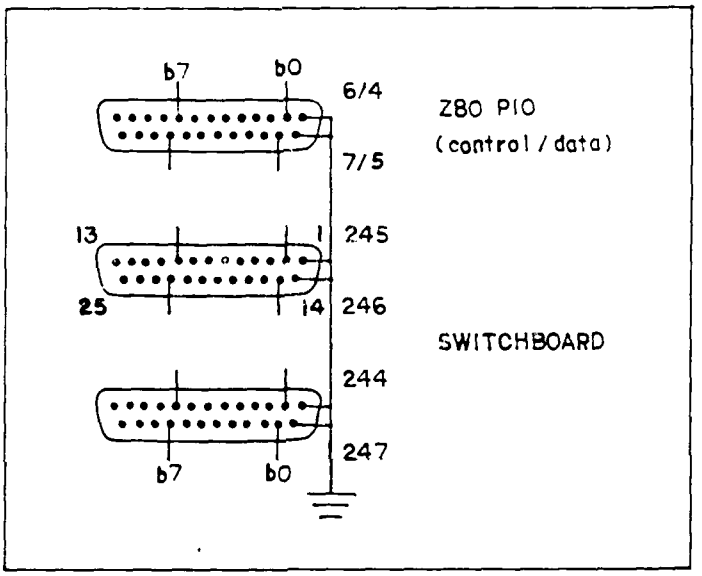


Figure IV-1. Parallel Port Configuration

Connections to the parallel ports of both the Switchboard and Super Quad board are made via dual row header pins. To provide easy access to the parallel ports, cables were constructed with the appropriate header on one end and female DB-25 connector on the other. Thus, the six parallel ports are available at 3 DB-25 connectors mounted on the rear panel of the mainframe. Figure IV-1 shows the location of each port at these connectors. This connection configuration was chosen because DB-25 mounting holes were

available and to provide compatibility with the author's modified Kaypro II microcomputer, which was used as a backup system. After all connections were made, each port was tested in both the input and output modes.

Software. The choice of a CP/M operating system was dictated by the author's familiarity with it, the desire to maintain compatibility with the backup computer, and the abundance of CP/M software available at AFIT. C was chosen for the primary programming language as a compromise between speed of execution and speed of program development. The BD Software C compiler version 1.5 was used exclusively. The speed optimization switch (-o) was used in compilation of all code to make it as fast as possible. (20:19) Because a floating point package for the C compiler was unavailable, Microsoft BASIC was used for number crunching programs as well as quick and dirty test programs. Efforts were made to use sound software engineering principles thoughout program development.

HP5340A Interface(6: Sec 2,14:96-97)

A Hewlett Packard model 5340A frequency counter was used for testing and calibration of the AFIT DCSO. This instrument was equipped with an HP-IB or IEEE-488 Standard interface port. When placed in the "Talk Always" mode via a switch on the HP5340A's back panel it continuously outputs data. This feature made interfacing the frequency counter to the computer a fairly straightforward task.

The IEEE-488 standard interface bus consists of 16 signal lines, functionally grouped into three component busses. An eight bit data bus transfers data in bit-parallel, byte serial form. Data transfer is controlled by a three wire handshaking or transfer bus. The remaining five lines constitute the general interface management bus. The latter bus is used by a controller to handle a number of instruments connected to the bus simultaneously. Since the primary concern for this case was merely reading the data provided by one instrument, the general interface management bus was not used.

Data is output by the HP5340A as a stream of ASCII characters in the following manner. The first character output is either a "D" or "L" indicating whether the measurement was made direct or using phase locked loops. The second character is either an "O" or space. The "O" indicates a display overflow. The third character is a space. Characters 4 through 14 contain the display digits in positive exponential form, most signigicant digit first. For instance, the character string "29899987E+1" is a valid series of display digits. Characters 15 and 16 are a carriage return and line feed, respectively. The ASCII code output by the 5340A is inverted, but this is quite easily remedied by reinverting the received character to its standard form within the interface software. Note also that the eighth data bit is not required for the transfer of the seven bit ASCII characters. The importance of this will be

apparent after discussing the HP5340A interface hardware.

The serial transfer of data bytes is controlled by the three handshake lines, Ready For Data (RFD), Data AVailable (DAV), and Data ACcepted (DAC). When the controller, in this case the microcomputer, is ready for data it indicates such by driving RFD high (>2.4 volts). Upon receipt of RFD high the HP5340A places the data byte on pins DO-D8. After a settling time it then sets DAV low (<0.4 volts). When the computer recognizes DAV low it is responsible for reading the data, setting RFD low and DAC high, acknowledging the transfer. The frequency counter then sets DAV high again. DAC can then be set low and the cycle repeated until all data is read. Details of handshake timing are shown in Figure IV-2a.

To perform the handshake data transfer sequence two output bits and one input bit are required in addition to the eight data lines. Although the bit-mapped port mode of the Z80 PIO can handle this situation easily, an alternative configuration allows the HP5340A interface to perform on any two ports (one input, one output). As previously noted the ASCII codes transferred do not require the eighth data bit. By simply leaving this bit unconnected and connecting the DAV line in its place on the input port the dependence on the bit-mapping capability of the Z80 PIO is eliminated. Appropriate software masking prevents interference of the DAV signal and data. The hardware connection thus shown in Figure IV-2c performed excellently.

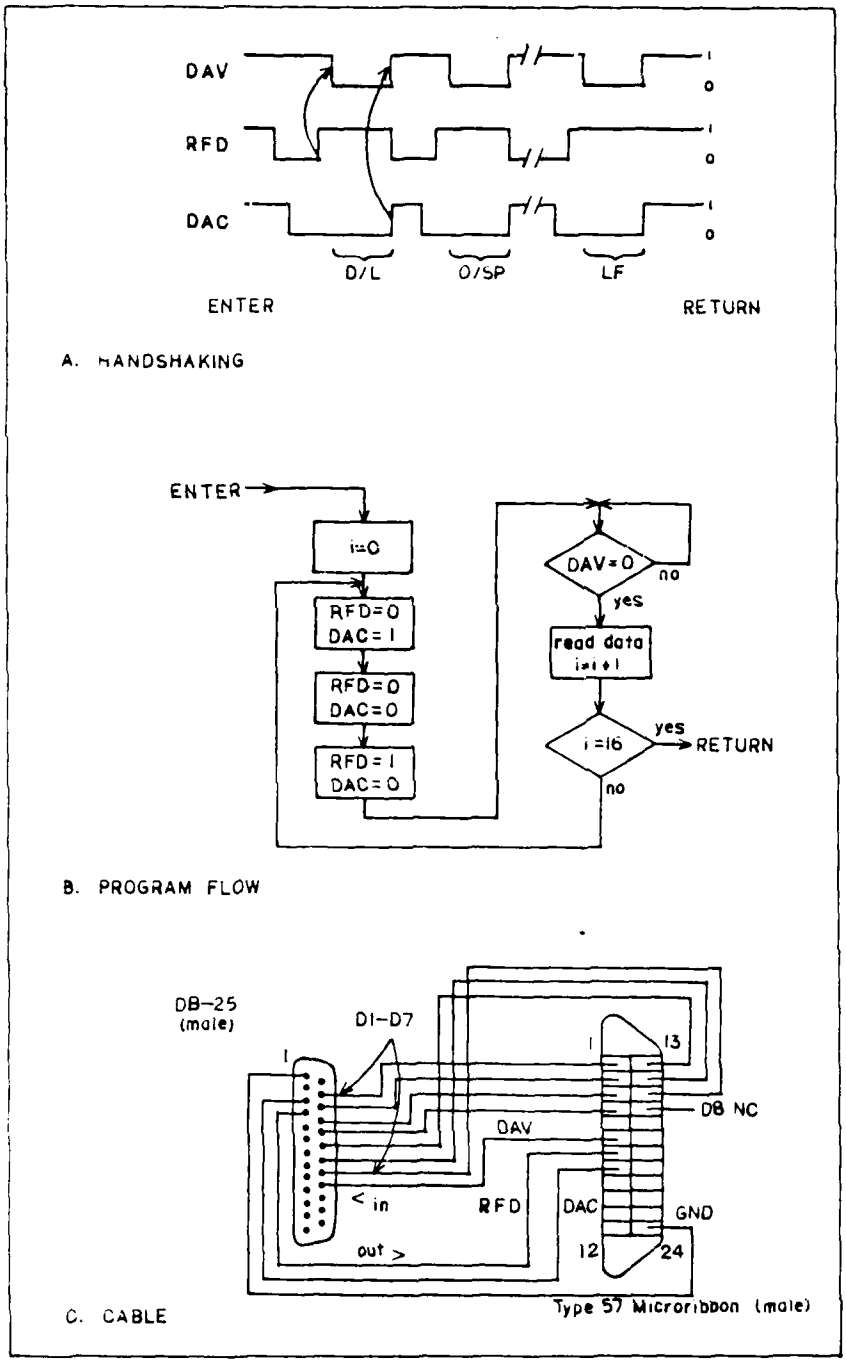


Figure IV-2. HP 5340A Interface Details

Figure IV-2b is a flow chart showing the operation of the frequency reading code (read_freq.c). When readings are not being taken, RFD and DAC are held high to allow the counter to continue running. Upon entrance to the module, RFD is set low, stopping the counter. This insures that the counter will be ready with the first data character. After setting DAC low also, RFD is set high initiating the data transfer. The DAV line is then monitored until it goes low. Then the data character is read and the character counter incremented. This process repeats until all 16 characters are read, but note that RFD and DAC are not allowed to both go high simutaneously until just before program control returns to the caller. This prevents the counter from running which would result in erroneous data readings. The read_freq.c code is included in Appendix C.

Cyborg Interface(3)

A Cyborg Thermal P642 electronic thermometer was used throughout temperature testing of the DCS, circuit. The Cyborg was chosen for its temperature resolution (0.01 degree), availability, and digital output capability. The P642 normally provides digital output as four BCD digits, at a rear panel mounted, DB-25 male connector. Reading this output would thus require two eight bit input ports. Unfortunately the manufacturer could not provide an accurate pinout of the connector. For these reasons, an alternative approach was taken.

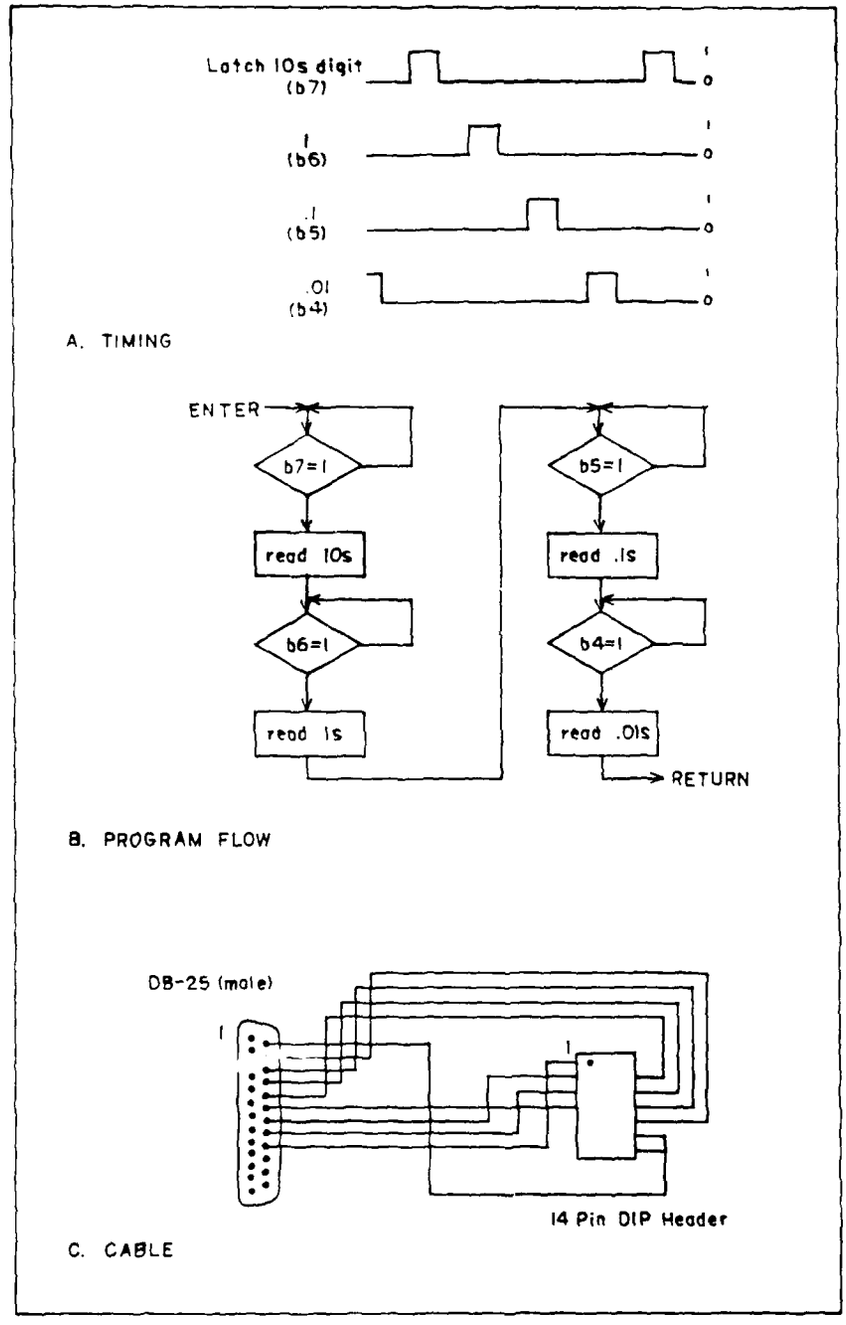


Figure IV-3. Cyborg Interface Details

Examination of the Cyborg's circuitry revealed that the BCD outputs are provided by four latches. After further inspection of the circuit it was discovered that the input to these latches consists of a four bit BCD bus, and four latch enable lines which are depicted in Figure IV-3a. These lines are provided at an internal 14 pin DIP socket. Through use of a 14 pin DIP header these lines can be connected to a single 8 bit parallel port (Figure IV-3c) and each BCD digit decoded by software.

The temperature reading module, read_temp.c, operates as shown in Figure IV-3b. Each latch enable line is monitored in turn, starting with that for the most significant digit. While each enable is high the corresponding digit is read, converted to the proper ASCII digit, and stored. After all four digits are read program control returns to the calling routine. The latch enable signals occur at a constant rate regardless of temperature change, and that rate is slow enough to allow accurate sampling by the computer.

V. Subsystem Testing

RF Circuitry Modification and Test

Several modifications to the DCSO radio frequency circuitry were required to make it operational. Upon initial startup it was found that neither of the SAW loops oscillated properly. Insertion loss and input impedance measurements led to the discovery of two broken wire bonds in the SAW device. After repair of the defective bonds, insertion loss and impedance were again measured and found to agree closely with McGuire's results. The tested SAW device was placed in the DCSO board but still the performance of both oscillator loops was unsatisfactory. Only after the following experimentally determined changes did both loops provide stable oscillation at the resonant frequencies reported by McGuire; 298 MHz for the clock loop, and 309 MHz for the thermometer loop (7: Sec III, 3-5). The capacitor in the clock loop was replaced with a one foot length of transmission line, by soldering two SMA connectors to the microstrip on each side of the former capacitor location. The 9 dB of attenuation was removed from the thermometer loop. In addition, an SMA connector was installed to allow measurement of the thermometer loop resonant frequency.

Plessey Prescaler Test

After insuring proper oscillation of both oscillator loops, the Plessey prescalers were functionally tested in the following manner. With power applied and the resets (pin 3) of both prescalers tied low, the TTL carry outputs (pin 11) were monitored with an oscilloscope. Neither prescaler operated correctly. To isolate the prescalers for further testing the attenuators between the power dividers and Plessey inputs were removed. A 300 MHz sinusoidal signal from a Wavetek frequency generator was then applied to the input of each prescaler in turn, while monitoring the output with an oscilloscope. By varying the input signal amplitude it was possible to get both prescalers working. The C-Chain prescaler operated with an input amplitude of approximately 400 mV peak-to-peak, but the T-Chain prescaler locked up with any input signal greater than 20 mV. According to the manufacturers specifications for the Plessey SP8735B an input signal of 400 to 800 mV peak-to-peak is required for proper operation (11:758). After much tedious troubleshooting, the problem was traced to a bad solder connection on the feedthrough which connects the bias decoupling capacitor to ground. After repairing this defect, proper operation of both prescalers was observed, with input voltage levels of up to about 600 mV peak-to-peak. A second attempt was made to observe correct (divide-by-eight) prescaling of the SAW oscillator frequencies after reinstalling the input attenuators. Again

this test failed. A second bout of troubleshooting led to the discovery that the input attenuators were reinstalled upside down (as they were originally), and provided no attenuation. Correct installation of the attenuators resulted in both Plessey prescalers working as designed.

The ability to stop the T-Chain was also tested. With a logic 0 on the inputs to the MECL translator (pins 6,7) the thermometer loop Plessey prescaler stopped counting and remained at the logic level it was stopped at. With the inputs to the MECL translator tied high (or floating high) counting was observed as described above.

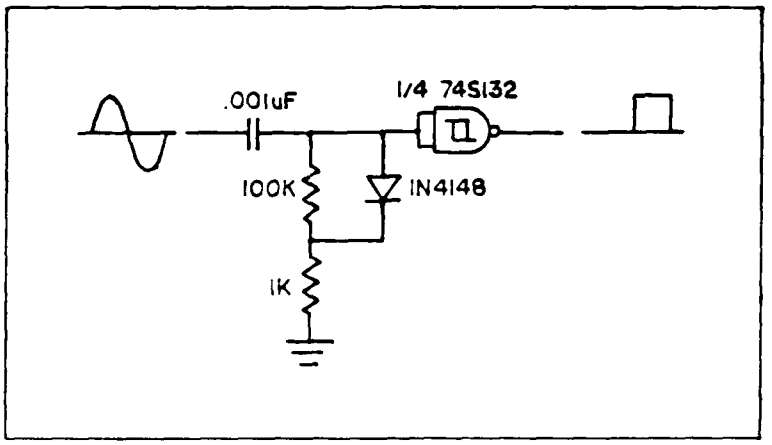


Figure V-1. 50 Ohm to TTL Interface Circuit

Thick Film Prescaler Test

Both in circuit and out of circuit tests were performed on the AFIT IC thick film prescalers. Using the thick film construction procedure described in Appendix D a test circuit minus the NMOS controller IC was fabricated and

mounted in the Elite solderless breadboard. The Elite board supplied power (pin 12) and ground (pins 18 and 40). A TTL level square wave input signal of 40MHz was supplied by the Wavetek in conjunction with the 50 Ohm to TTL Interface circuit shown in Figure V-1. This diode clamp circuit, used by Slobodnik in the original DCSO work, converts a high frequency sinusoidal input signal to a TTL level square wave (16: 28). Figure V-2 shows both the input signal from the clamp circuit and the resulting output from the thick film prescaler as measured at the C-osc wire bonding pad. Although the signals shown appear noisy, the noise can be attributed to the test setup, and the photo clearly shows divide-by-ten operation. Testing of the T-Chain thick film prescaler yeilded similar results. Resetting of both prescalers was also successfully checked, by grounding the appropriate reset pad. When placed in the DCSO circuit board, both thick film prescalers performed as designed. Indeed, as shown by Figure V-3 the same C-Chain prescaler's output while in the DCSO is a fairly clean 4 volt square wave. Using the HP5340A frequency counter, the original oscillator frequencies as well as the prescaled signals were measured and compared, as shown in Table V-1. Percent difference calculations on these results yeild less than one-thousandth percent difference indicating that all digital prescaling circuitry is fully functional.

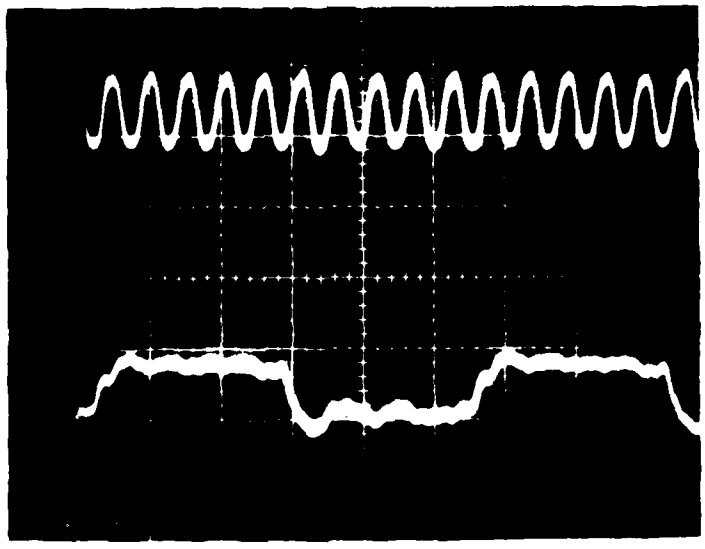


Figure V-2. Out of Circuit Thick Film TTL Prescaler Test
(Vertical: 5 V/div. Horizontal: 0.05 uS/div.)

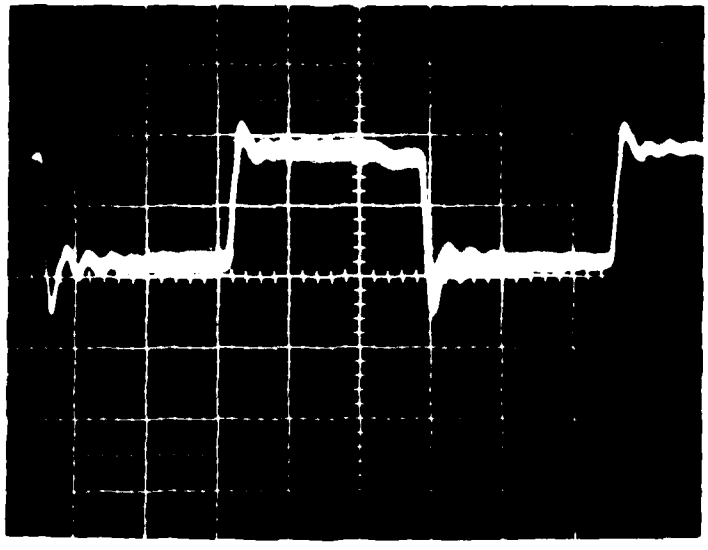


Figure V-3. In Circuit Thick Film TTL Prescaler Test
(Vertical: 2 V/div. Horizontal: 0.05 uS/div.)

Table V-1. Prescaler Test

	Oscillator frequency (MHz)	Oscillator frequency /80 (MHz)	Measured prescaled output (MHz)
Clock loop	298.90127	3.73626	3.73627
Thermometer loop	309.75573	3.87194	3.87185

NMOS DCSO Controller Chip Testing

Three versions of the NMOS DCSO controller integrated circuit were received from MOSIS. Upon receipt, each chip was labeled with a letter series designation and chip number. An early version of dcso_ver1.00, submitted before completion of ESIM, arrived first and was designated the "A" series. Thirteen A series chips were packaged and seven unpackaged. A more fully simulated (using ESIM) dcso_ver1.00 batch arrived later and was labled series B. The C series controller chips, were fabricated on the same processing run as the B chips using dcso_ver1.10. As mentioned in the chip design chapter, the dcso_ver1.10 chips include a one-shot circuit on the C-chain resets. Eleven packaged and no unbonded chips were received for both the B and C series.

Prior to functional testing, the controller chips were microscopically inspected and power consumption measurements made. The next section begins with a discussion of these observations. Then follows a description of the computer test setup developed to exercise the chips. Testing of each major subsystem of the controller chip is then detailed. In general, test results apply to all chips except where

noted. A summary of NMOS contoller chip performance concludes the chapter.

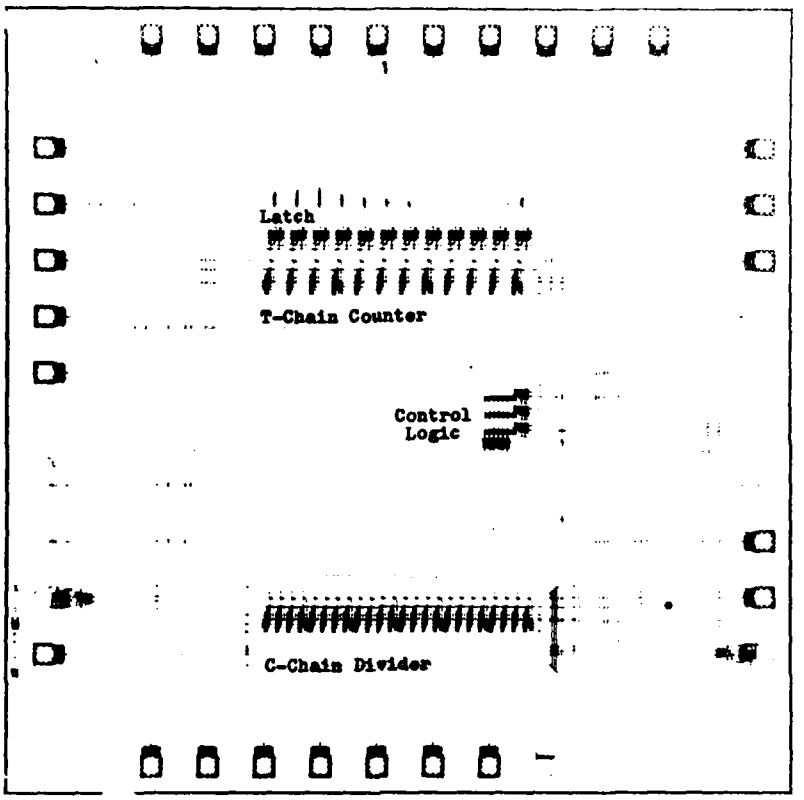


Figure V-4. Photomicrograph of NMOS Controller Chip (dcso_ver1.00(B))

Inspection and Power Test. Before testing, two chips were randomly chosen from each batch for microscopic inspection. No fabrication errors were visible. One design error was evident on the first dcso_ver1.00(A). The location of this error is marked by an asterisk on Figure V-4, which is a photomicrograph of dcso_ver1.00(B). On series A chips the marked conductor is diffusion instead of metal causing a transistor to be formed by the unbuffered clock line that crosses over.

Power testing was also conducted by random sampling. Three chips were chosen from each batch, and connected to Vdd=5volts (pins 24,25) and Ground (pins 30,31). An ammeter in the Vdd line provided current consumption measurements. During power testing the substrate connection (pin 1) was also grounded. For all chips current draw was measured to be between 60 and 80 milliamps. To insure the relatively high current was not due to grounding of the buffered output pad also connected to pin one, the pin 1 to pad 1 bond wire was removed from each of the test chips and the power test repeated. No difference in current consumption was observed. Using 70 milliamps as a mean current value the average power consumption is easily calculated to be (70 mA)(5v)=350mW. Throughout controller chip testing, the current of test chips was monitored to provide an early warning of chip malfunction. No such cases were observed.

Computer Test Setup. The circuit shown in Figure V-5 provides the computer to controller chip interfacing required for automated testing. This circuit was constructed on an Elite solderless breadboard for ease of development and flexibility in use. The bit-mapping feature of the Z80 PIO is required for the A port which has three outputs and five inputs. Switches and jumpers in the C-osc and T-osc lines allow selection of the clock signal source. The clock can thus be strobed by the computer, Elite board clock, or Wavetek signal generator. The Wavetek was used in conjunction with the 50 Ohm to TTL interface circuit (Figure

V-1) to supply clock signals of greater than 1 MHz, the limit of the Elite board's internal clock. The Elite board supplies a variable clock signal (up to 1 MHz) as well as power and ground.

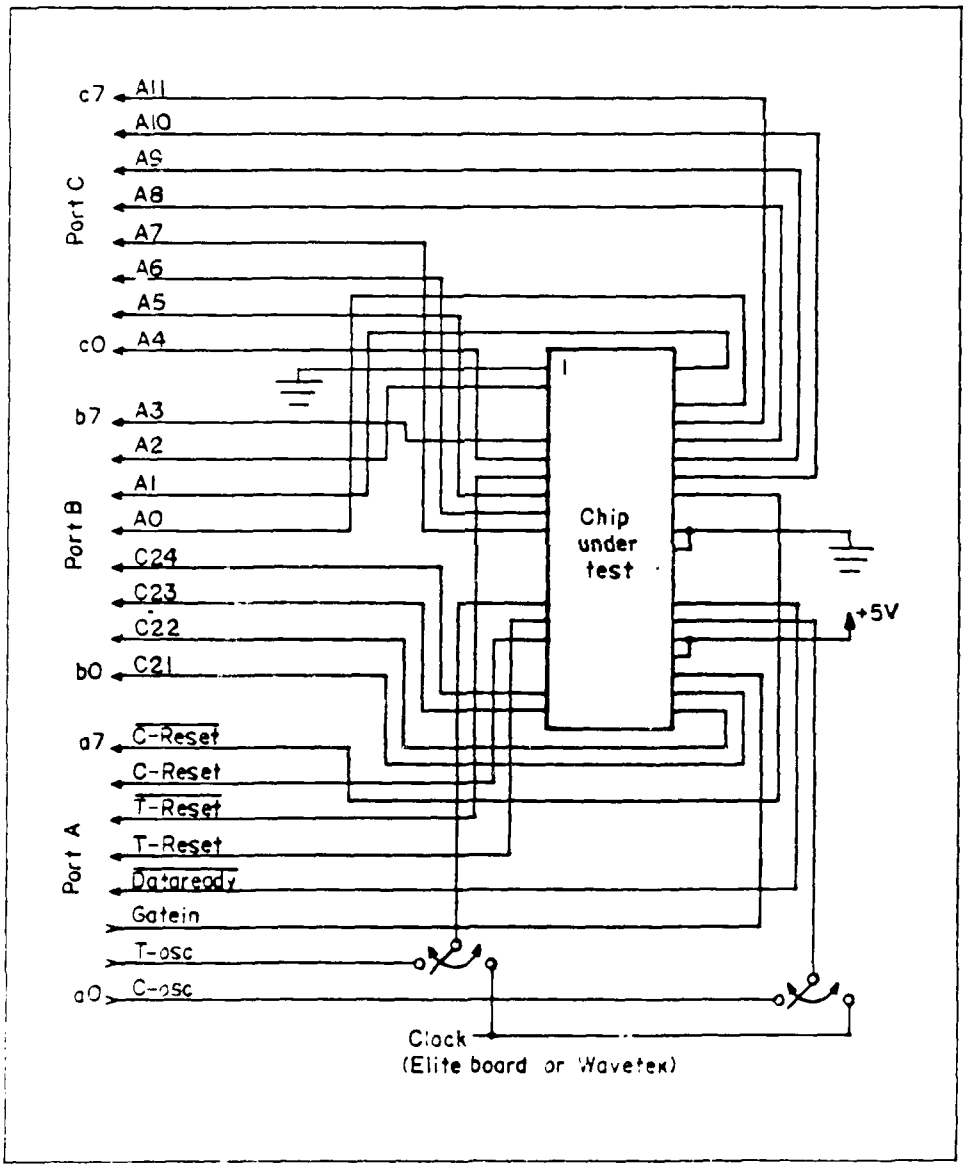


Figure V-5. Chip Test Computer Setup

A variety of software was written for manipulation of the controller chip inputs and monitoring of outputs. In lieu of a detailed discussion of the software which can be found in Appendix C, the computers role as a test instrument will be discussed in the sections that follow.

C-Chain Test. Several tests were performed to verify proper operation of the C-Chain divider. First, a 4MHz signal was applied to C-osc and the first bit of the C-Chain (C1-pin 18) monitored with an oscilloscope. This test proved that the C-chain was being properly clocked. As this is also the fastest changing bit the test also showed that the C-chain meets speed requirements. Secondly, the computer was used to strobe C-osc while also monitoring C21-C24 in an effort to count the number of pulses required to turn over each of the bits. Results of this test were fairly random and the test prohibitively long, due to the length of the divider chain and computer's relatively slow strobing. The PIO ports were later found to be malfunctioning and may have contributed to the erroneous results. However, this test did show that the C21 output (at pin 19) of the A series chips never changed from the low logic level. A third test proved the C-Chain divider to be operating correctly. While supplying a 256.410 KHz C-Osc signal, the time between transitions of bits C22-C24 was measured using an oscilloscope and manual stop watch. Several data readings were taken and averaged for each bit to compensate for human error and possible frequency

fluctuations. This information allows simple calculation of both the theoretical and actual frequency at each bit. As summarized in Table V-2, results of this somewhat crude test were very accurate. While performing the above test it was discovered that the C-Chain outputs had a small step from 0 to about 1 volt while at the logic 0 level. This glitch was observed on all chips tested, and appeared to be the result of switching in the previous counter stage.

Table V-2. C-Chain Test Results

Bit	C22	C23	C24
Half period length (Sec.)	4.10 4.09 4.03 4.13	8.33 8.19 8.14 8.22	16.39 16.46 16.45 16.44
Mean	4.08	8.22	16.43
Frequency (Hz)	0.12	0.06	0.03
$\frac{\text{Input Freq. (Hz)}}{2^{(\text{Bit\#} - 1)}}$	0.12	0.06	0.03

To test the functionality of the C-Chain at operating speed, a similar test was performed with an input frequency of about 4 MHz, while using the computer to monitor C22-C24. Instead of a real time clock, a software counter measured the time between changes in the outputs. Results of the fourth test showed the C-Chain to be operating properly at speed. During C-Chain testing the Gatein pin (23) was grounded to prevent C-reset signals from occurring. Despite this the C-Chain appeared to reset whenever the C-osc signal

was stopped. No explanation for this phenomenon was found. Further discussion of desired resetting is presented in the control logic test section.

T-Chain and Latch Test. Testing of the T-Chain counter proceeded in much the same manner as for the C-Chain, but was hampered greatly by lack of a Dataready input pad. Experience proves that such capability is required for testing. The Dataready signal, which is generated by the control logic, latches the T-Chain count, as shown in the chip design chapter. It was assumed during chip design that Dataready could be set by selectively stopping the C-Chain while manipulating Gatein, but this proved to be impossible.

First, as with the C-Chain, T1 was checked using an oscilloscope and 4 MHz input signal, and T-Chain clocking verified. Next, while monitoring the address outputs by computer the entire chip was run with the same clock on C-Osc and T-Osc and Gatein wired to C23. Theoretically under these conditions, the latched output should be all zeros except for a logic 1 on A1, because the T-Chain should look exactly like the lower 12 bits of the C-Chain. This test revealed that data was in fact being latched, but did not produce a constant address output as expected. Attempts to run the C and T chains at different speeds also failed to produce consistent addresses. Because the C-Chain could not be stopped with the Dataready line high as planned during design, a check of T-Chain counting was impossible by

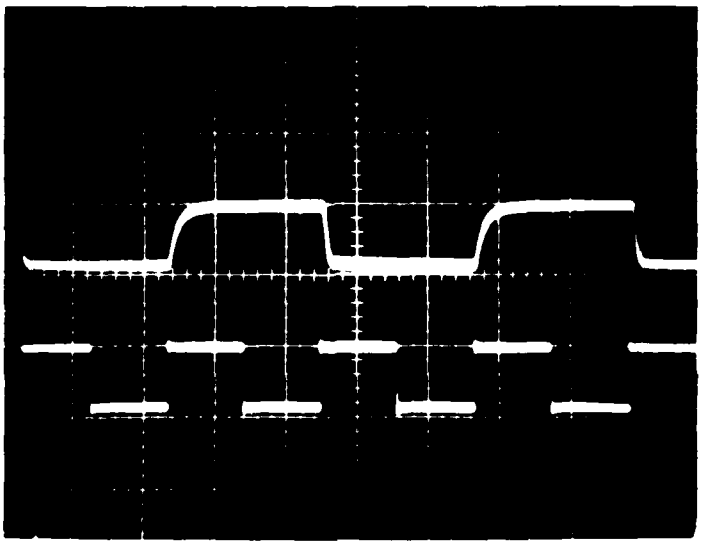


Figure V-6. A0 (top) vs T-osc (bottom)
(Vertical: 5 V/div. Horizontal: 5 uS/div.)

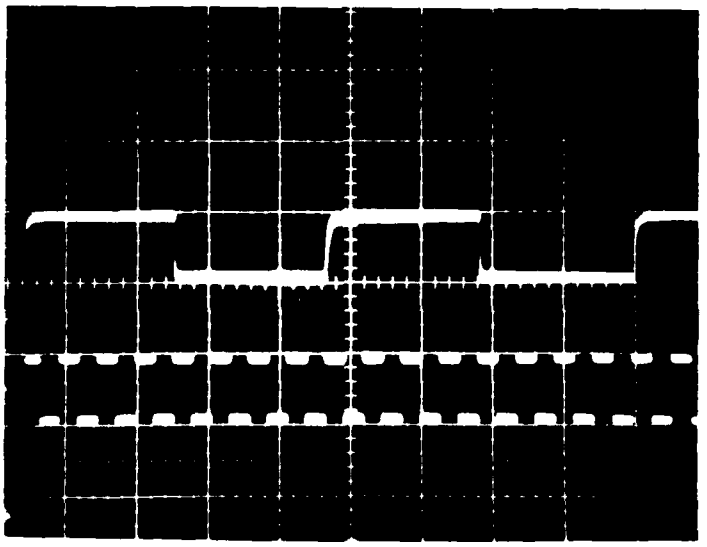


Figure V-7. A3 (top) vs T-osc (bottom)
(Vertical: 5 V/div. Horizontal: 20 uS/div.)

external manipulation of the appropriate signals. Thus, another approach to testing of the T-Chain was devised.

Mounting a small solderless breadboard on the platform of a Wentworth Labs probe station allowed direct manipulation of the DataReady and T-Reset lines via careful probing of the metal conductors. To insure contact, two probes were used on each run, and continuity between probes checked with an ohmmeter before applying input signals. With DataReady held high and T-Reset low, proper counting of the T-Chain was observed as shown in Figures V-6 and 7. Proper counting was also evident at higher stages of the counter. A high level on the T-Reset line forced all outputs low, as expected. All measurements made while probing the controller chips were performed in darkness, as any light, expecially that of the probe station microscope, caused random clocking signals on all address outputs.

Control Logic Testing. It was assumed in design of the controller chips that the C-Chain could be stopped at various points to allow testing of the control logic. Unfortunately that assumption proved invalid. While running the C-Chain (at 4MHz) with Gatein tied high, or connected to one of the C-Chain outputs it was possible to "see" the control logic signals occurring with a pulse catching logic probe. But this method did not allow verification of signal timing. Attempts to view the control logic signals with an oscilloscope failed, because the signals were inconsistent. Thus, an attempt was made to verify signal timing by running

the C-Chain, hence the control logic more slowly and observing the available control signals by computer. The results of this test were also inconclusive as sometimes the control signals were detected and sometimes not. In retrospect, this test was probably faulty due to a failing PIO in the computer. A probe station test like that described for the T-Chain was also performed on the control logic. Using probes to manipulate the inputs to the control logic, and another probe connected to the oscilloscope to test each output, the control logic appears to operate as designed. Thus, a second check of the control logic while running the C-Chain at a lower frequency is suggested.

Summary of Controller Chip Tests. The tests performed indicate that both the T-Chain and C-Chain count properly for input signals up to slightly over 4 MHz, but cannot be run very slowly or stopped. Glitches, of about 1 volt were noted in the C-Chain outputs while at the low logic level, and are apparently caused by switching in the previous counter stage. When properly strobed, the T-Chain latches hold data. The Control Logic, though it may indeed perform as designed statically and at low speed, is clearly inadequate at the 4 MHz operating speed required for the controller chip. Possible causes for this problem may be excessively long input lines, improper NAND gate design or most likely, trying to run the control logic too fast. It was also discovered during testing of the DCSO Plessey prescalers that an inverted Gate signal (active low) must be

supplied. On chip provision for generating this signal was overlooked during chip design. Due to these failings of the controller chips, the one-shot circuit of dcso_ver1.10 was not tested.

VI. DCSO Calibration and Test

To allow calibration and testing of the AFIT DCSO, off-board control circuitry was developed. This logic utilizes the AFIT IC thick film prescalers and the C-Chain of the NMOS controller chips. Additional circuitry, required to implement the control logic and latched T-Chain, is mounted on two SK-10 solderless breadboards. This approach was chosen, instead of scrapping the thick film and NMOS chip entirely, to demonstrate the feasibility of the thick film AFIT IC approach, and provide information for future controller IC work. After development of working control circuitry the DCSO was calibrated and tested over a 0 to 70 °C temperature range.

Control Circuitry

The working DCSO control circuitry is shown in block diagram form by Figure VI-1. Details of the off-board control circuitry are provided by Figure VI-2. Although the T-Chain and latch are quite similar to those designed for the NMOS implementation, with the exception of five additional counter stages; the control logic differs extensively. This is because the C-Chain divider of the NMOS controller chip cannot be reset from an external source, thus the C-Chain runs continuously.

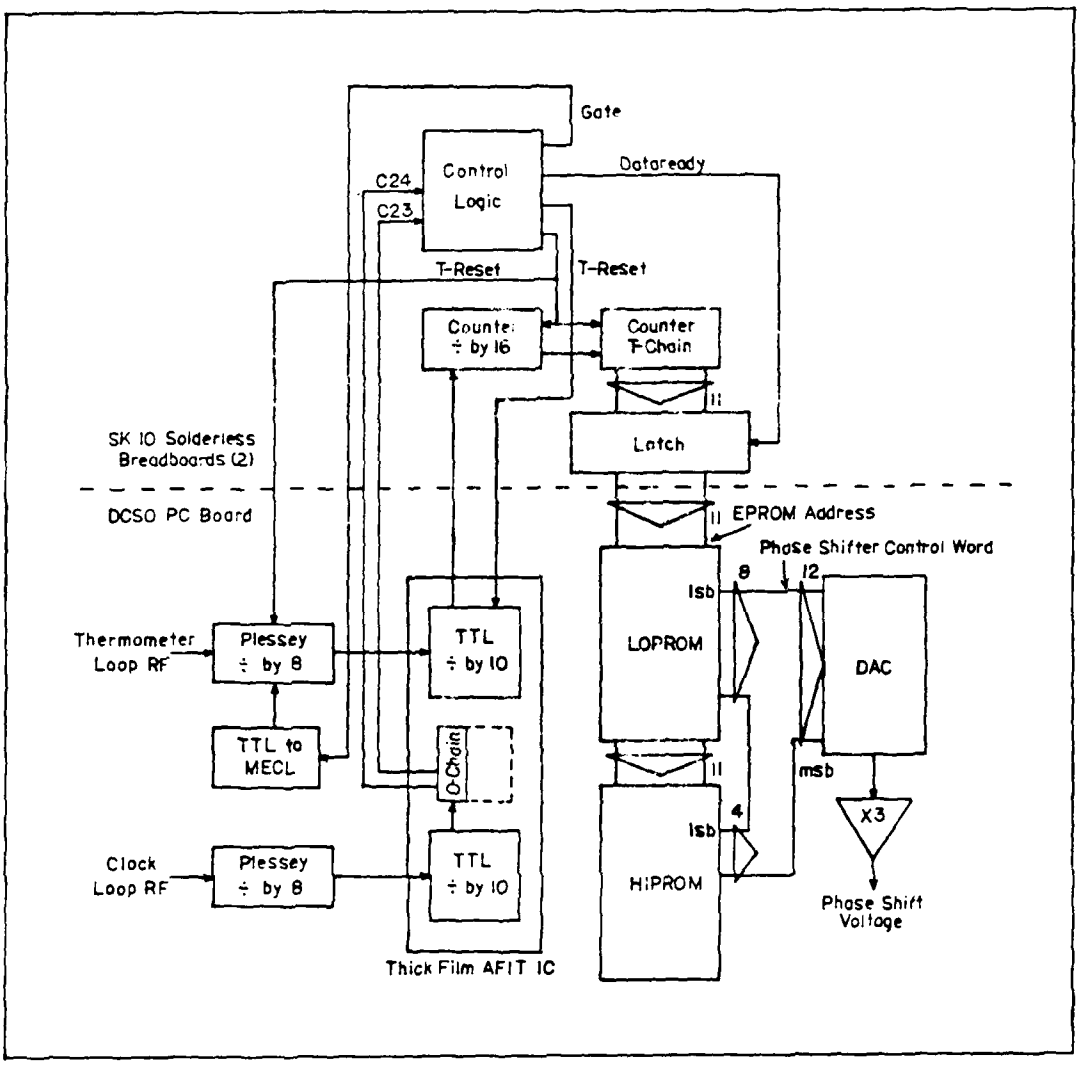


Figure VI-1. DCSO Control Circuitry Block Diagram

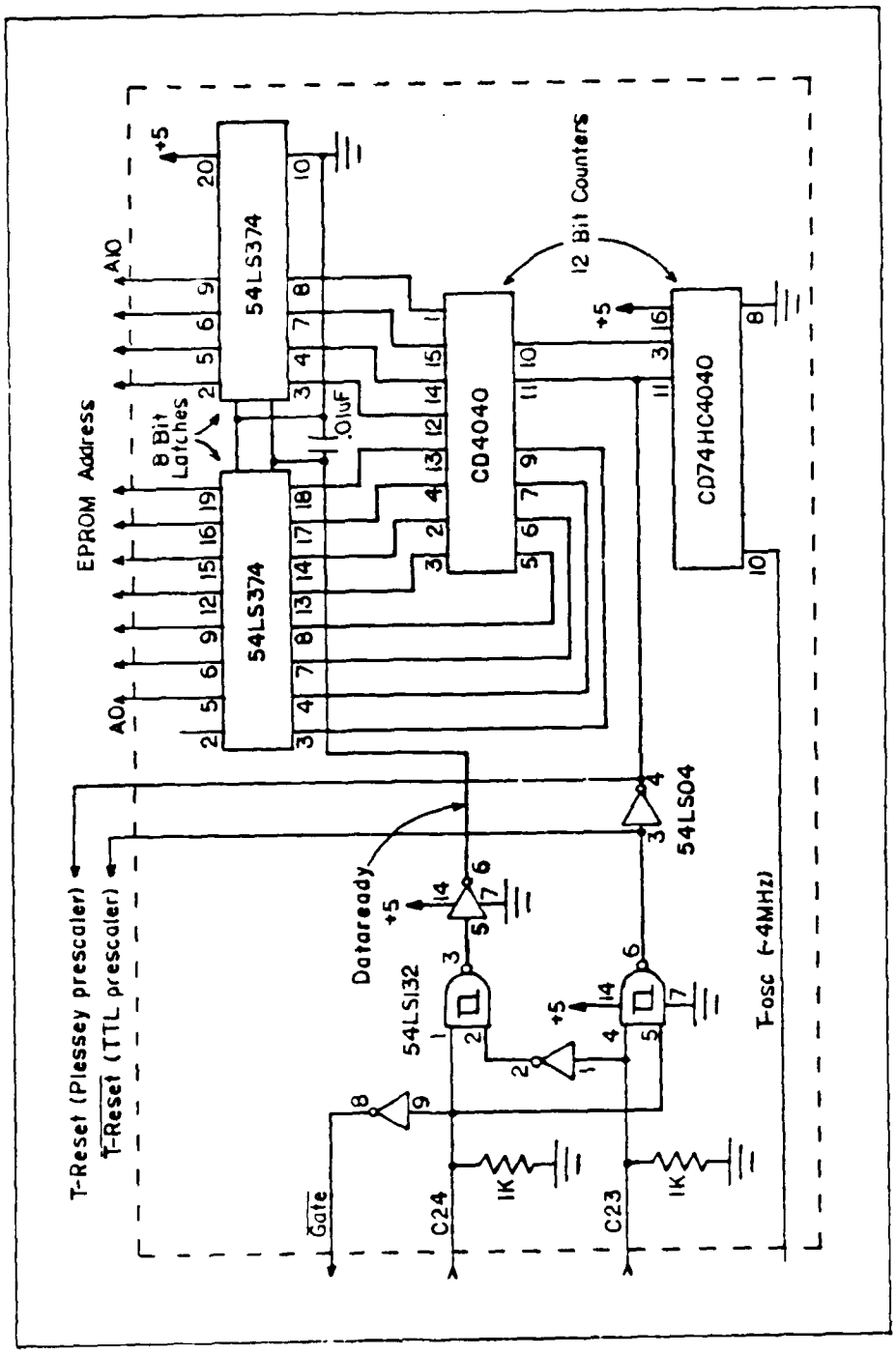


Figure VI-2. Breadboarded Control Circuitry Schematic

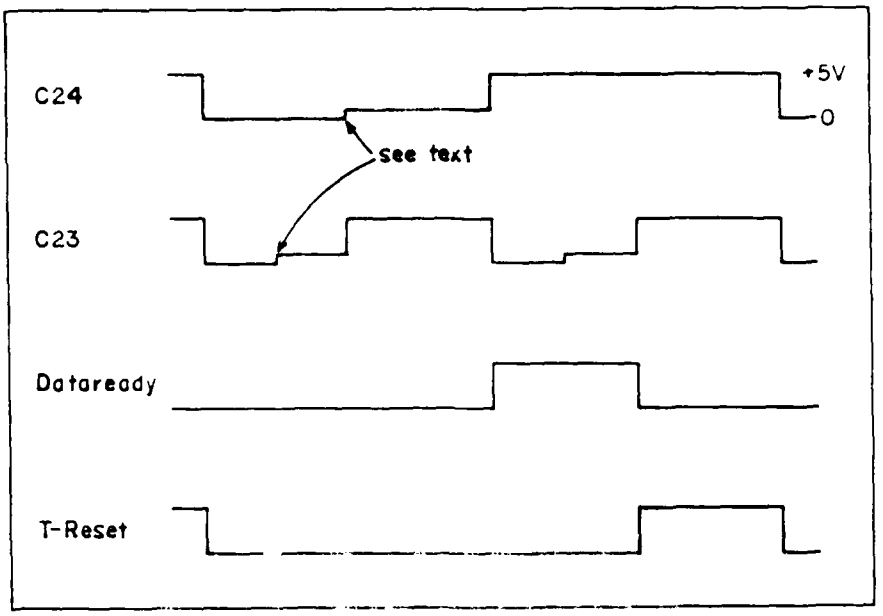


Figure VI-3. Control Circuitry Timing Diagram

Figure VI-3 is a timing diagram for the working control logic. While C24 is low the T-Chain counts. When C24 goes high, the T-Chain is stopped via the T-Chain Plessey prescaler clock inhibit which is driven by the MECL translator. With C24 high and C23 low, Dataready goes high, latching the T-Chain count. When C23 goes high, Dataready is removed, and the T-Reset signals are generated. Counting resumes when C24 returns low. With the continuously running, non-resettable C-Chain there is significant "dead time", because the T-Chain counts only when C24 is low. Thus, with quickly varying temperature the compensation circuitry may not be able to keep up. Due to the inability to reset the C-Chain, the degradation of system performance is unavoidable.

The 54LS132 NAND gates decode C23 and C24 to provide the necessary control logic signals. The Schmitt triggered inputs of these gates in conjunction with the 1K loading resistors compensate for the glitch in the logic low levels of C23 and C24 noted previously. A .01 microfarad capacitor on the DataReady line prevents the 54LS374 Octal D-Type (positive edge triggered) latches from triggering on spurious noise which is a problem otherwise. The CMOS 4040 type counters were chosen because they provide a resettable 12 bit binary ripple counter in a 16 pin package. The CD74HC4040 part has a typical maximum frequency rating of 50 MHz while the CD4040 is limited to about 3.5 MHz with a 5 volt power supply (12:123; 13:106). Data sheets for the integrated circuits used in construction of the breadboarded control logic are provided in Appendix E. The additional prescaling of the T-osc signal was found to be necessary after an initial calibration attempt showed that the T-Chain counts repeated three times over the desired temperature range. Presumably this could have also been corrected by shortening the counting interval (i.e. using bits C21 and C22 to drive the control logic), but circumstances prevented such an attempt. Recall that C21 did not work on the A series NMOS chips, the only version for which unbonded chips were received. Also, the wire bonding pad for C22 was damaged during construction of the test thick film.

The test thick film was constructed using the procedure described in Appendix D. In addition to the bonds required

for operating the C-Chain, two jumper bonds connecting thick films pads 14 to 13 and 11 to 7 were made. The first jumper conducts the prescaled T-Osc signal to pin 39 of the thick film, while the latter allows the T-Chain TTL (thick film) prescaler to be reset via pin 37. The test thick film is soldered into a 40 pin IC socket for additional mechanical strength. The pins of the socket are bent slightly to fit in the non-standard 40 pin socket on the DCSO board. Thirty guage wire-wrap wire soldered directly to the thick film pins (Berg clips) provide the necessary connections to the breadboard circuitry.

The addresses generated by the breadboard T-Chain are returned to the DCSO board via wire-wrap wires to a specially made 24 pin socket for one of the EPROMs. Also included on the breadboard (for testing), but not shown in Figure VI-3, are LEDs on the address lines, and wiring to allow computer monitoring of those lines. The address LEDs provide a means of visually verifying circuit operation during test setup. Despite its appearance, the control circuitry works well.

Calibration

Figure VI-4 shows the DCSO calibration setup. While running the DCSO in an environmental chamber, the computer monitors the clock loop frequency and T-Chain counts (EPROM addresses), while holding the clock loop frequency stable via the digital phase shifter circuitry. With the clock

loop frequency held constant, the latched T-Chain counts provide a valid measure of crystal temperature. When both the T-Chain count and clock loop frequency are stable a valid data point is known.

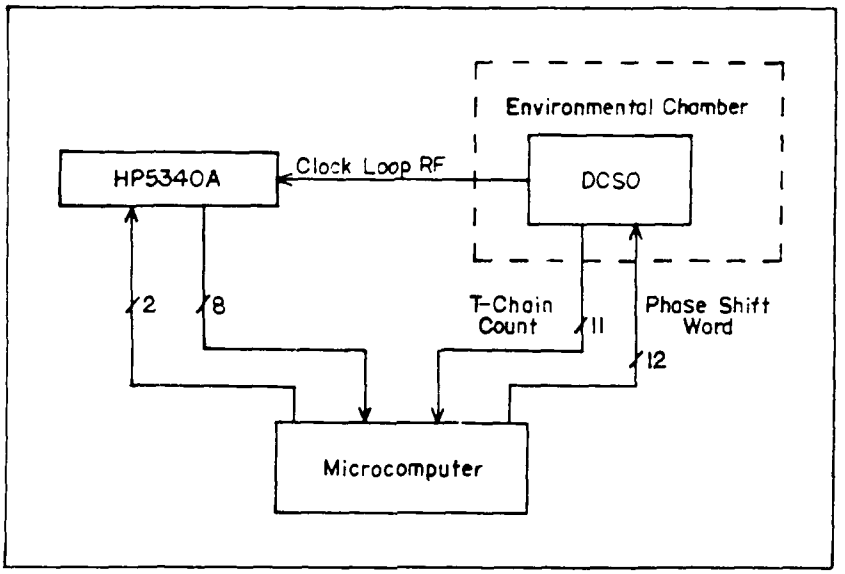


Figure VI-4. Calibration Setup

A flow diagram for the calibration program DCSOCAL.BAS is shown in figure VI-5. Basic was chosen for this program to shorten development time and allow for easy program changes. The calibration program reads the clock loop frequency from the HP5340A via the previously described interface. If the measured frequency is less than the desired frequency, the phase shifter control word is decremented until the frequency falls within a specified range. When the measured frequency is too high the or site occurs. Prior to calibration, the desired frequen was chosen to be midway between the maximum and minimum

frequencies obtainable by manipulating the phase shifter control word at the end points of the temperature range. The clock loop frequency, deviation from desired frequency, phase shifter control word, and EPROM address are displayed for each cycle on the computer's CRT. When the displayed values have stabilized, data is recorded manually.

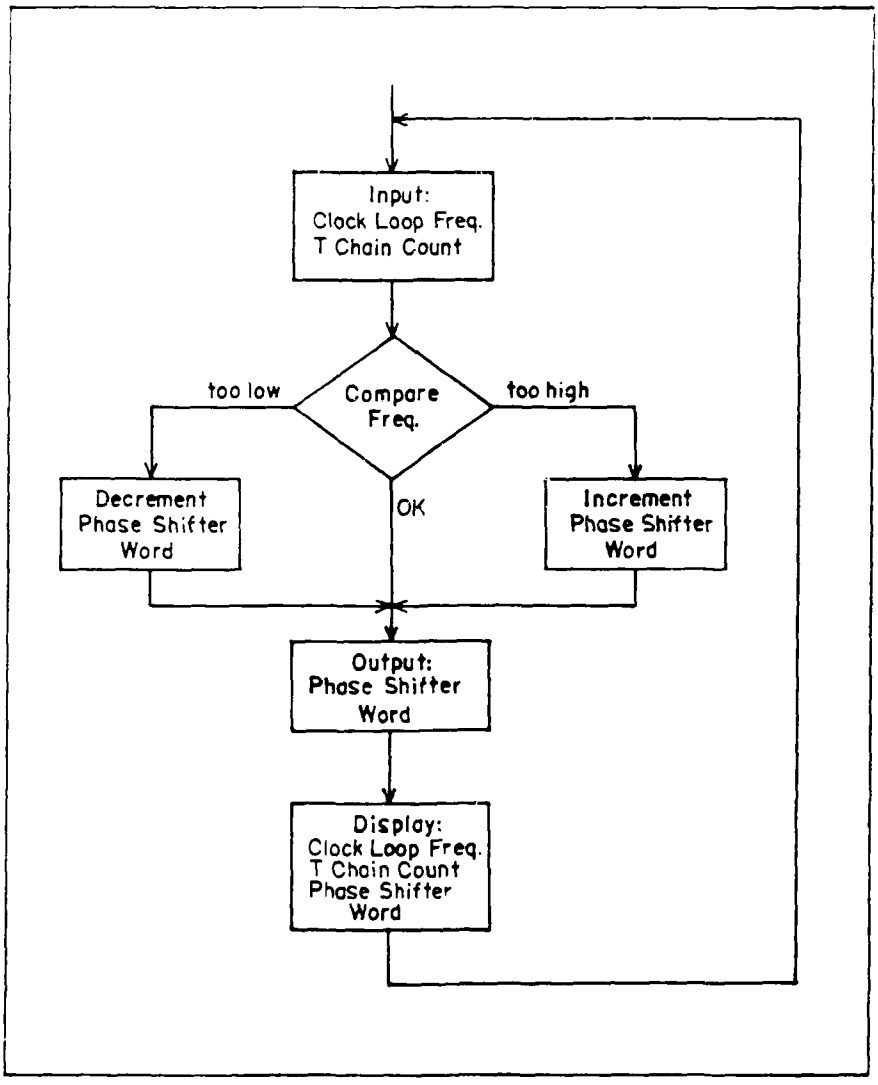


Figure VI-5. DCSO Calibration Program Flow

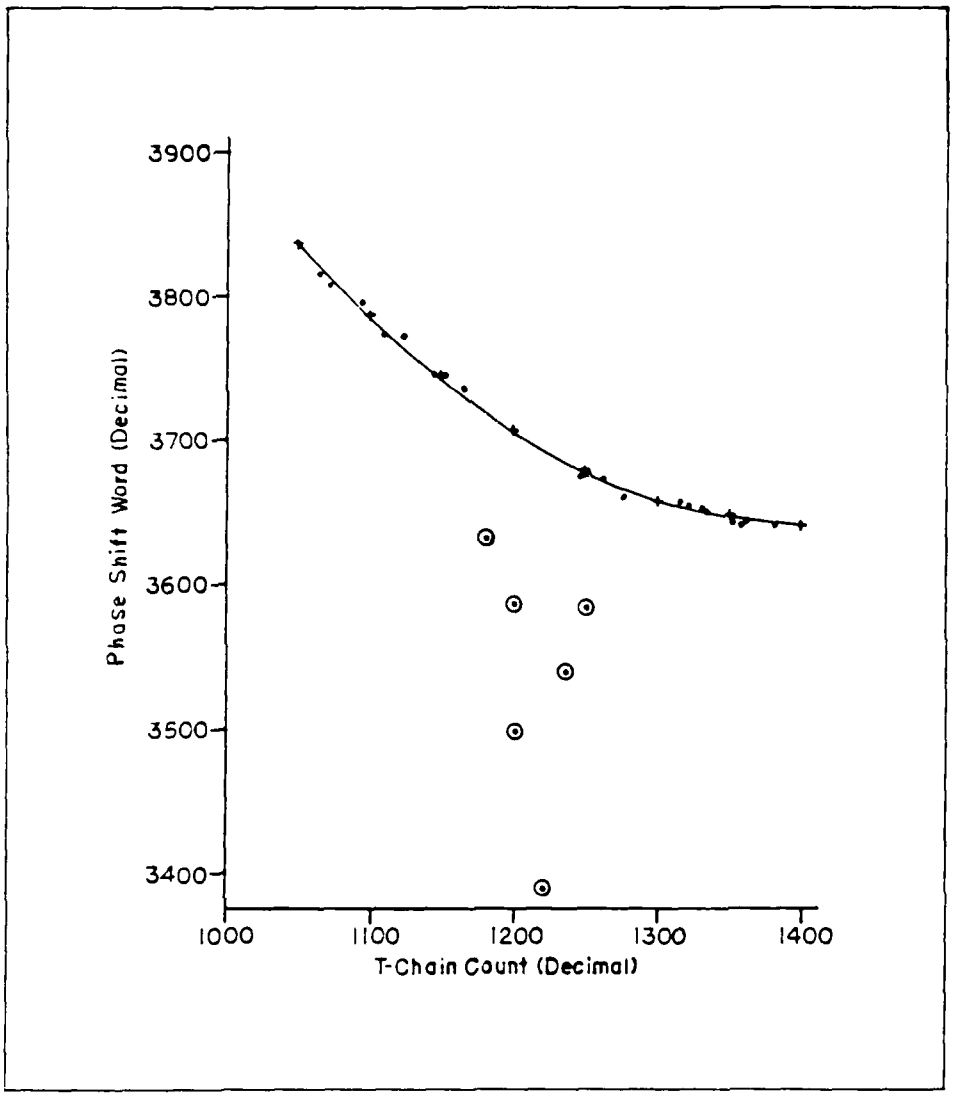


Figure VI-6. Calibration Data

Data points were taken throughout the temperature range by manually adjusting the setting of the environmental chamber, and allowing each point to settle for about 10 minutes. Temperature was monitored using the Cyborg electronic thermometer. Calibration data is provided in Appendix F and shown graphically in Figure VI-6. With the exception of six points, the calibration data forms a smooth curve. In the excepted region it was impossible to hold the clock loop frequency stable (even with increased settling time), and the data points represent the least unstable readings.

The MBASIC program CALCURVE.BAS (see Appendix C) takes as input the calibration data points and generates two EPROM files in the following manner. First, a second order curve fit is performed on the data points by the least squares method. The algorithm parallels a textbook example problem and is constrained to a second order curve fit (19:156). The equation produced by the curve fitting portion of the program is then used to generate phase shifter control words for all EPROM addresses in the calibration range. Constants are generated for EPROM addresses outside the calibration range. The two 2716 EPROMs are wired as a 2 K by 16 bit memory, as shown in Figure VI-1, but only a 12 bit phase shifter control word is required by the DAC (12 bits). Thus, the EPROM labeled LOPROM contains the low order eight bits of the phase shifter control words, and HIPROM contains the high order four bits (stored as the low order bits).

The low order eight bits and high order four bits of the phase shifter control words are stored in the files LOPROM.MAC and HIPROM.MAC respectively, as they are generated. When written to the files, both portions of the phase shifter control word are written as a byte in hexadecimal notation, but each entry in HIPROM.MAC has a leading zero (i.e. 0E hexadecimal). For each entry in both files, the address is included as a comment. The format of the EPROM files is such that they can be directly assembled with the MACRO-80 assembler. Using LINK-80, with code origin at 0000H, .COM files are produced for EPROM programming. Both EPROMs were programmed and verified using a Cyclic Redundancy Check (CRC). For this calibration the CRCs were found to be 7228 hex and 156E hex, for HIPROM and LOPROM respectively.

For the test results shown in the following section, the unstable data points were omitted from consideration, and the curve drawn in Figure VI-6 represents the curve fit without these points. Clearly, the cause for this instability needs to be discovered and corrected before a truly good calibration can be accomplished. In an attempt to determine the cause of the frequency instability in the 30 to 50 °C range, the outputs of the clock loop and thermometer loop were monitored with a spectrum analyzer while varying the temperature from 0 to 70 °C. In the excepted region, the entire frequency spectrum of the clock loop exhibited an amplitude drop of about 8 dB. The

thermometer loop showed no anomalies.

DCSO Test Procedure

Despite the suspect calibration data, EPROMs were programmed and the DCSO tested over a 0 to 70°C temperature range. Figure VI-7 shows how the computer was used to monitor temperature and clock loop frequency. Two test programs were written for this test setup. The program dcsotest.c reads the Cyborg and automatically displays data every 2 degrees. The program takedata.c is similar but records data only when a key on the computer console is pressed, thus permitting data to be taken after a timed period of stabilization. Both programs are run under CP/M with the control P option to provide hardcopy data, and are provided in Appendix C.

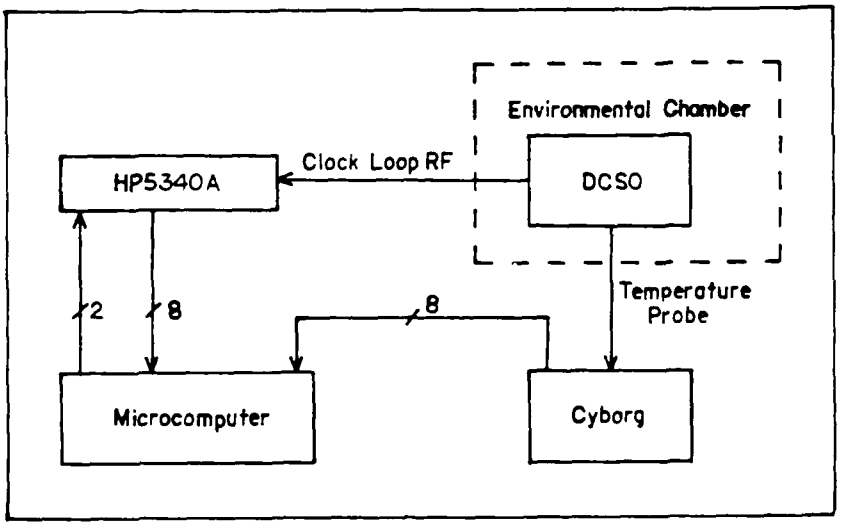


Figure VI-7. DCSO Test Setup

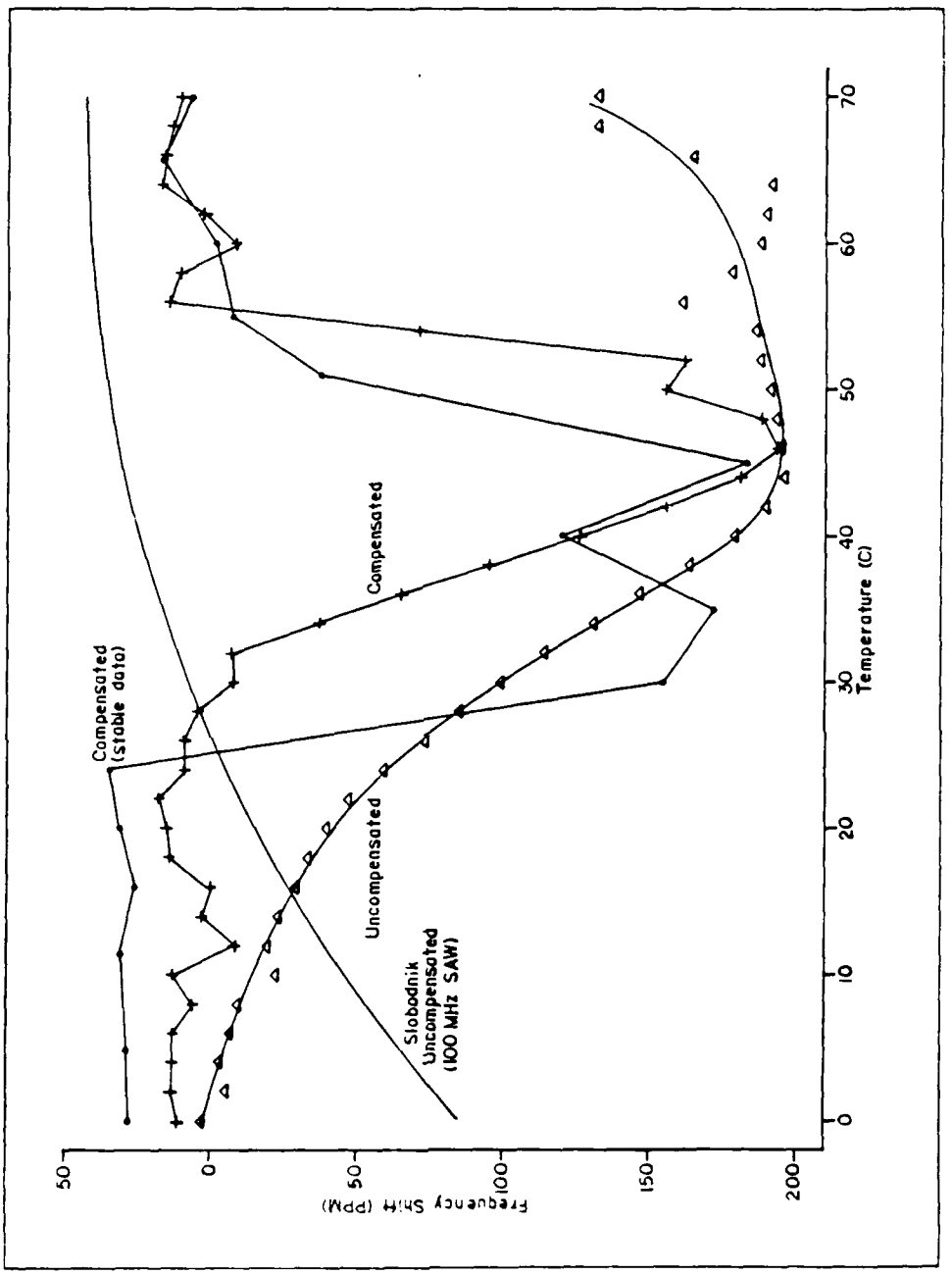


Figure VI-8. DCSO Test Results

Prior to testing of the DCSO, the uncompensated circuit was tested to provide a baseline measure of frequency drift versus temperature. Without the calibration EPROMs installed, the DCSO board was placed in the environmental chamber and cooled to 0 °C while running. After both temperature and clock loop frequency stabilized the test began. The program dcsotest was started with printer output, and the oven setting changed to about 75 C. As the oven heated, the computer automatically recorded frequency data, "on the fly", every 2 degrees Centigrade. After installing the calibrated EPROMs, the same procedure was repeated. Also, because the time response of the DCSO was expected to be poor, another test was performed. For this test the DCSO was cooled to 0 °C and the temperature increased in approximately 5 °C increments. Data was recorded using takedata for manual triggering of data recording after each point stabilized for about 15 minutes.

DCSO Test Results

Results of the DCSO system test are shown graphically in Figure VI-8 as frequency shift in parts per million versus temperature. In the 0 to 70 °C range, Slobodnik's data for a 100 MHz SAW device clearly shows that frequency increases with increasing temperature (16:26). The uncompensated curve shown in Figure VI-8 shows that the frequency of the AFIT DCSO decreases with increasing temperature over most of this range. This discrepancy leads

to the conclusion that the frequency dependence shown is not that of the SAW device. Although these results are far from optimal, much useful information regarding AFIT DCSO performance can be gleaned after some explanation.

Both of the compensated curves show that the compensation circuitry in fact works. This is most evident in the 0 to 25 °C region. The large frequency shifts measured in the 30 to 50 °C region resulted from the omission of the unstable calibration data, and erratic behavior in this region. At higher temperatures compensation is again visible. Repeatability of the compensation circuitry is shown by the relative shape of the compensated curves. Reproducible data is virtually impossible with the SAW package not hermetically sealed, because SAW devices are affected by pressure and humidity as well as temperature. Such effects also explain why the compensated data (0 to 25 C), is greater than the desired frequency, and why the desired frequency is not somewhere midway between the maximum and minimum points of the uncompensated curve. Because both the uncompensated curve and compensated-increasing temperature curves lag behind the compensated-stable temperature curve, it can be assumed that the SAW crystal responds to temperature changes more slowly than does the Cyborg. As the temperature probe was simply placed under the SAW device package, such an observation is not unexpected. Comparison of the two compensated curves over the 0 to 25 °C region provides a relative measure of the

systems time response, or tracking. As it took about 20 minutes to heat the oven from 0 to 70 °C, the rate of temperature change was approximately 3.5 degrees/minute. Even with this fairly large temperature gradient the compensation circuitry held the frequency stable within less than 30 parts per million in the 0 to 25°C range.

VII. Conclusions and Recommendations

Conclusions

Although this project did not produce a fully integrated DCSO, several accomplishments toward that goal can be cited.

1. After making several changes to the RF circuitry, both oscillator loops of the DCSO were made operational. The resonant frequencies of 298 MHz for the clock loop and 309 MHz for the thermometer loop were then verified.

2. Proper divide-by-eight operation of both Plessey high speed prescalers was verified, after installing the input attenuators (to the Plesseys) properly. A bad ground on the thermometer loop Plessey prescaler bias decoupling capacitor was also corrected to make it functional.

3. An additional TTL prescaler was added to further divide each of the oscillator frequencies by ten, making the highest frequency seen by the NMOS control logic approximately 4 MHz. This was accomplished by making the AFIT IC a thick film hybrid circuit consisting of two commercial TTL prescalers and an NMOS controller IC. The operating thick film circuit (including NMOS controller chip) was cycled over a 0 to 70°C temperature range several times without failure, indicating that the thick film hybrid AFIT IC approach is feasible.

4. An NMOS controller integrated circuit was designed and fabricated using lamda=2 micron technology. Testing of the controller chip revealed that while portions of the chip worked, the chip as a whole failed to meet design criteria.

5. Due to the failure of the NMOS controller chip to operate properly, off-board control circuitry was developed to allow calibration and testing of the DCSO as a system. Calibration and testing were facilitated by development of a computer system with interfaces to a Hewlett Packard 5340A frequency counter and Cyborg electronic thermometer.

6. The clock loop frequency was found to be unstable in the 30 to 50°C temperature range during calibration. A calibration curve was generated, with the unstable points omitted, and calibrated EPROMs programmed. The DCSO was tested over a 0 to 70°C temperature range. Test results show that the clock loop frequency does not increase with increasing temperature, indicating the temperature dependence observed was not solely a function of the SAW device. Despite this observation, the compensation circuitry was shown to compensate the frequency for the temperature induced variation in the 0 to 25°C range.

Recommendations

Though this thesis effort advanced the development of the AFIT single circuit board DCSO implementation considerably, several tasks remain before completion of the project. Specific recommendations for future work are

provided in this section.

1. The cause of the non-SAWlike behavior and instability of the clock loop must be discovered and corrected. One clue is the 8dB decrease in signal amplitude which occurs in the 30 to 50°C temperature range, indicating a possible temperature related failure of a component (most likely active) or connection in the loop. Because of the problems experienced with broken wire bonds in the SAW device and the RF circuit modifications required to get the oscillator loops working, the SAW is not above suspicion. Thus, the SAW device should first be removed from the DCSO and tested over the desired frequency range to insure that it exhibits the proper frequency versus temperature characteristics. This test should include both resonant frequency and amplitude measurements for each temperature point. After verification of the SAW device the operation of both oscillator loops should again be checked under varying temperature conditions (with the phase shifter voltage held constant at about 15 volts). If the unstable behavior of the clock loop is still evident, all components and connections in that loop should be systematically checked, beginning with the Avantek MICamp (because it is the only active component in the loop). If the unstable behavior of the clock loop is not apparent while testing with a constant phase shifter voltage, the phase shifter voltage should be changed via the DAC inputs to try to reproduce the unstable behavior. Stability may be further

checked using the calibration setup.

2. A second iteration of the NMOS controller integrated circuit should be developed using the knowledge gained in this work as a guide. The Cnt and CntRestore standard library cells should not be used for the counter/divider chains because they cannot be stopped without losing the count data. A flip-flop type counter cell should be used instead. The latch cells used for this iteration of the controller chips worked well, given proper control signals. Provisions for inputting all internally generated control signals should be included for test purposes.

3. To provide further information on the length of the counting chains required, it might prove beneficial to develop a test board using commercial counter ICs such as the 74L5196 and CD 4040 type chips in standard packages. This board could ride piggy-back on the DCSO and plug into the 40 pin AFIT IC socket. Such a board would allow the length of the counter and divider chains to be experimentally determined for most efficient use of the EPROM memory, and optimal system performance.

Bibliography

1. Advanced Digital Corporation. Technical Manual for SuperQuad.

2. Bahl, I.J. and Trivedi, D.K. "A Designer's Guide to Microstrip," Microwaves, 16 (5): 174-176+ (May 1977).

3. Cyborg Corporation. P642 Feedback Thermometer Handbook. Boston: 1977

4. Hahn, Danny and Elam, Frank. VLSI Counter and Latch. EE6.95 Report, 8 June 1983.

5. Hahn, Danny and Elam, Frank. DCSO Counter/Latch Testing. EE6.96 Report, 16 September 1983.

6. Hewlett Packard. HP5340A Frequency Counter Manual.

7. McGuire, Jerry W. Single Circuit Board Implementation of a Digitally Compensated SAW Oscillator (DCSO). MS Thesis. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, December 1983. (AFIT/GE/EE/83D-44).

8. Mead, Carver and Conway, Lynn. Introduction to VLSI Systems. Reading, Massachusetts: Addison-Wesley Publishing Company, 1980.

9. Morrow Designs. Switchboard User's Manual. 1979.

10. Newkirk, John and Mathews, Robert. The VLSI Designer's Library. Reading, Massachusetts: Addison-Wesley, 1983.

11. Plessey Solid State. Integrated Circuits Databook. Irvine, California: 1983.

12. RCA Solid State. CMOS Integrated Circuits Databook. USA: 1983.

13. RCA Solid State. QMOS High-Speed CMOS Logic ICs. USA: 1983.

14. Ricci, David W. and Nelson, Gerald E. "Standard Instrument Interface Simplifies System Design," Electronics, 95-106 (November 14, 1974).

15. Sims, G.R., Varnum, B.R., Hall, N.D., Staubs, P.G., Carter, H.W., and Colvin, R.D. LSI Implementation of a Digitally Compensated SAW Oscillator (DCSO) Controller Circuit. Interim Report. Wright-Patterson AFB, Ohio: Air Force Institute of Technology (1982).

16. Slobodnik, A.J., Jr., Colvin, R.D., Roberts, G.A., and Silva, J.H. Design and Test of a Digitally Compensated SAW Oscillator (DCSO). In-House Report. Griffiss AFB, New York: Rome Air Development Center, June 1981. (RADC-TR-81-200).

17. Texas Instruments Incorporated. The TTL Data Book for Design Engineers. USA: 1976.

18. Texas Instruments Incorporated. The TTL Pocket Data Book. USA: 1983.

19. Wylie, C. Ray. Advanced Engineering Mathematics. New York: McGraw-Hill Book Company, 1975.

20. Zolman, Leor. BDSC Compiler V1.5 User's Guide. Brighton, Massachusetts: BD Software.

Appendix A: Microstrip Calculation

Calculation of the width of the microstrip lines used on the AFIT IC thick film hybrid circuit was performed using the design equations provided in "A Designer's Guide to Microstrip" (2), as follows.

Alumina substrate parameters: thickness = 0.025 inch = h
relative dielectric = 9.6 = Er

Desired impedance: Zo = 50 ohms

Design Equations (2):

$W/h = [8 \exp(A)]/[\exp(2A) - 2]$, for $W/h <= 2$

$A = Zo/60\sqrt{(Er+1)/2} + [(Er-1)/(Er+1)][0.23+(0.11/Er)]$

neglect dispersion below;

$f(GHz) = 0.3\sqrt{Zo/(h\sqrt{Er-1})}$, h in cm

Calculations:

$A = 50/60\sqrt{10.6/2} + [8.6/10.6][0.23+0.11/9.6] = 2.11$

$W/h = [8 \exp(2.11)]/[\exp(4.22)-2] = 0.995$, <= 2

checking dispersion;

$f(GHz) = 0.3\sqrt{50/(0.0635\sqrt{8.6})} = 4.9$ GHz (no problem)

Thus,

$W = 0.995h = (0.995)(0.025'') = 0.0248$ ''

W 0.025 , so use 1/4 inch tape on times 10 scale artwork.

Appendix B: Chip Design Details

This appendix contains four cifplots showing how the major functional blocks of the DCSO controller IC are implemented in NMOS, the files used for simulating the chip (ESIM files) together with the output of each simulation, and a sample DRC output.

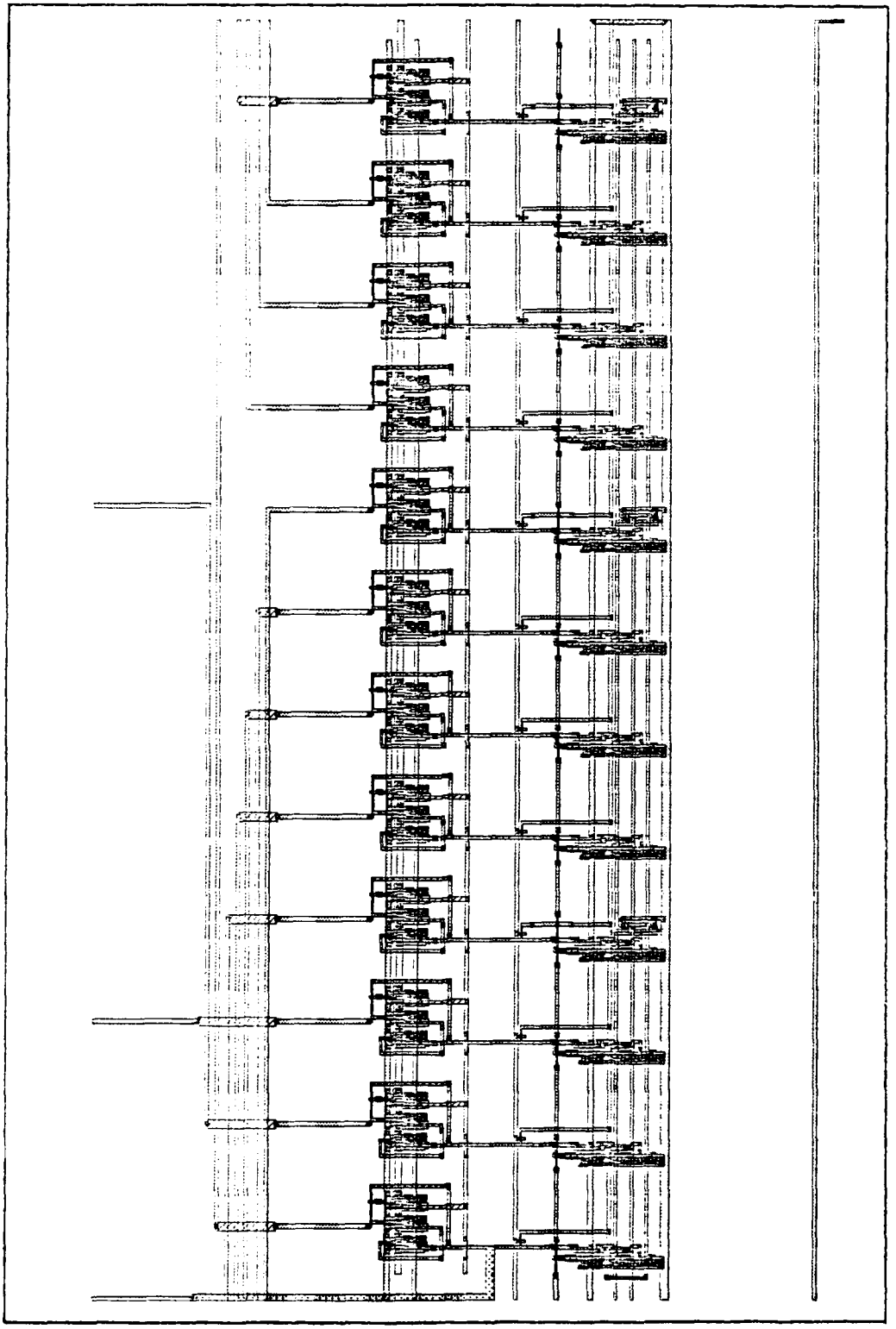


Figure B-1. T-Chain Cifplot

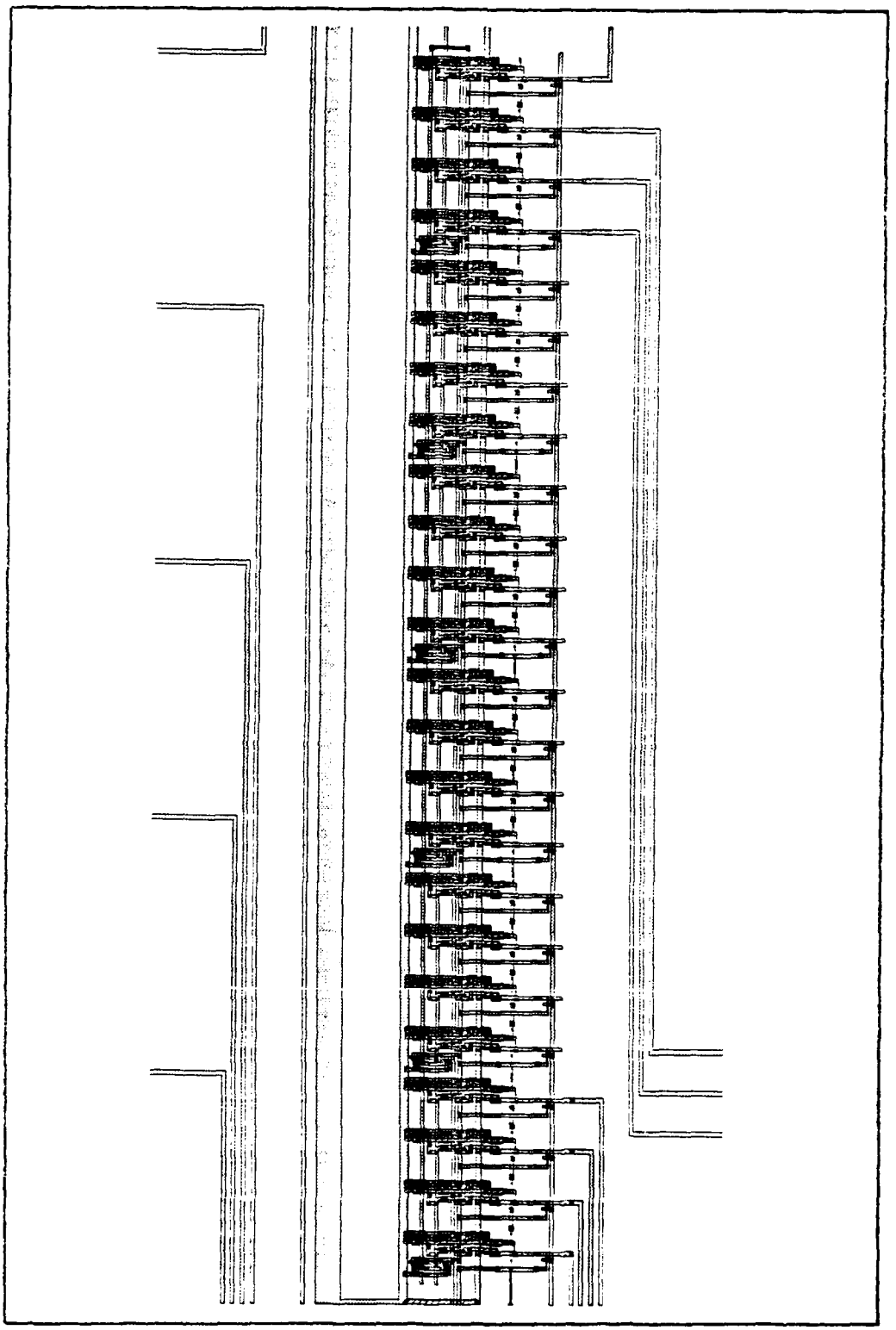


Figure B-2. C-Chain Cifplot

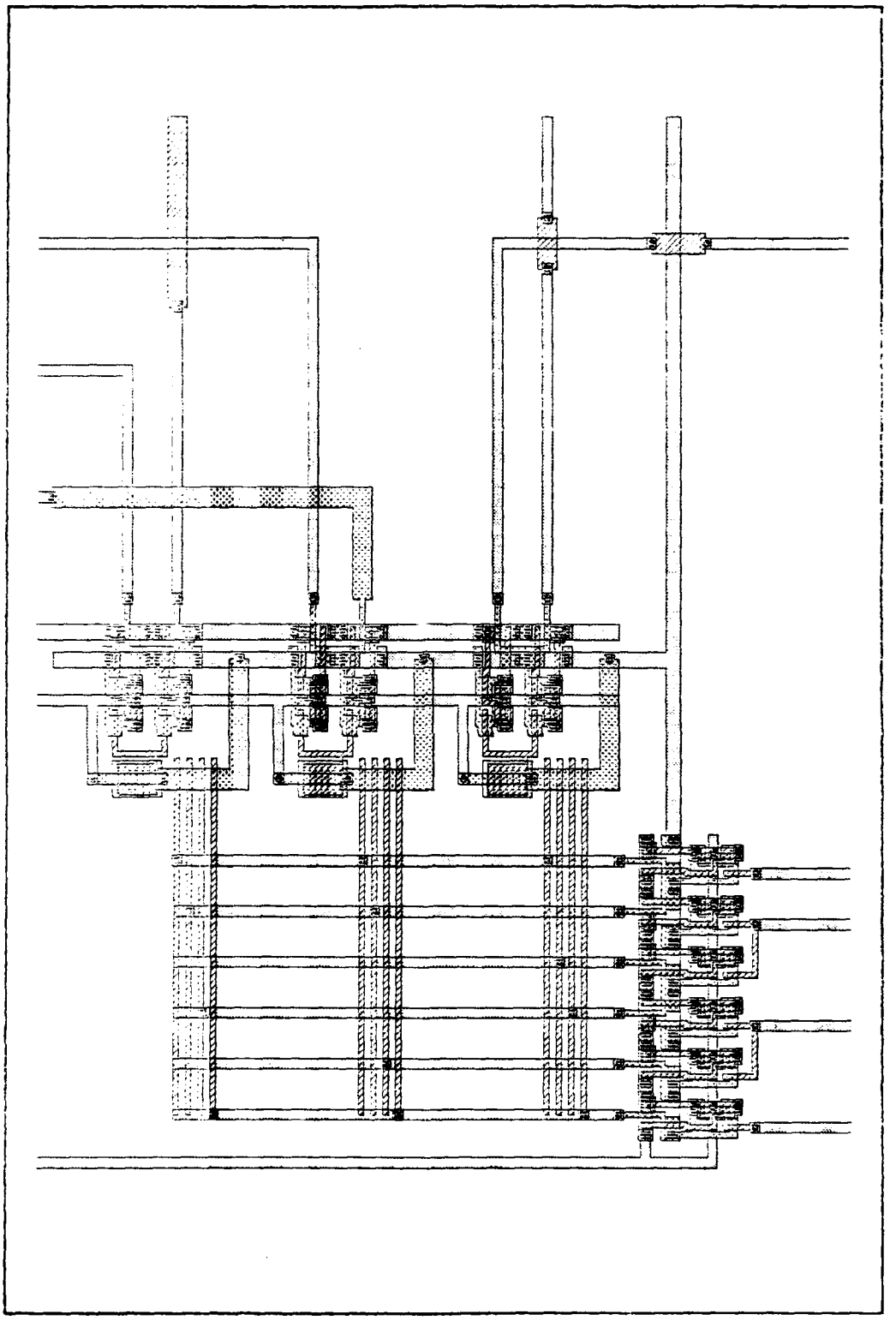


Figure B-3. Control Logic Cifplot

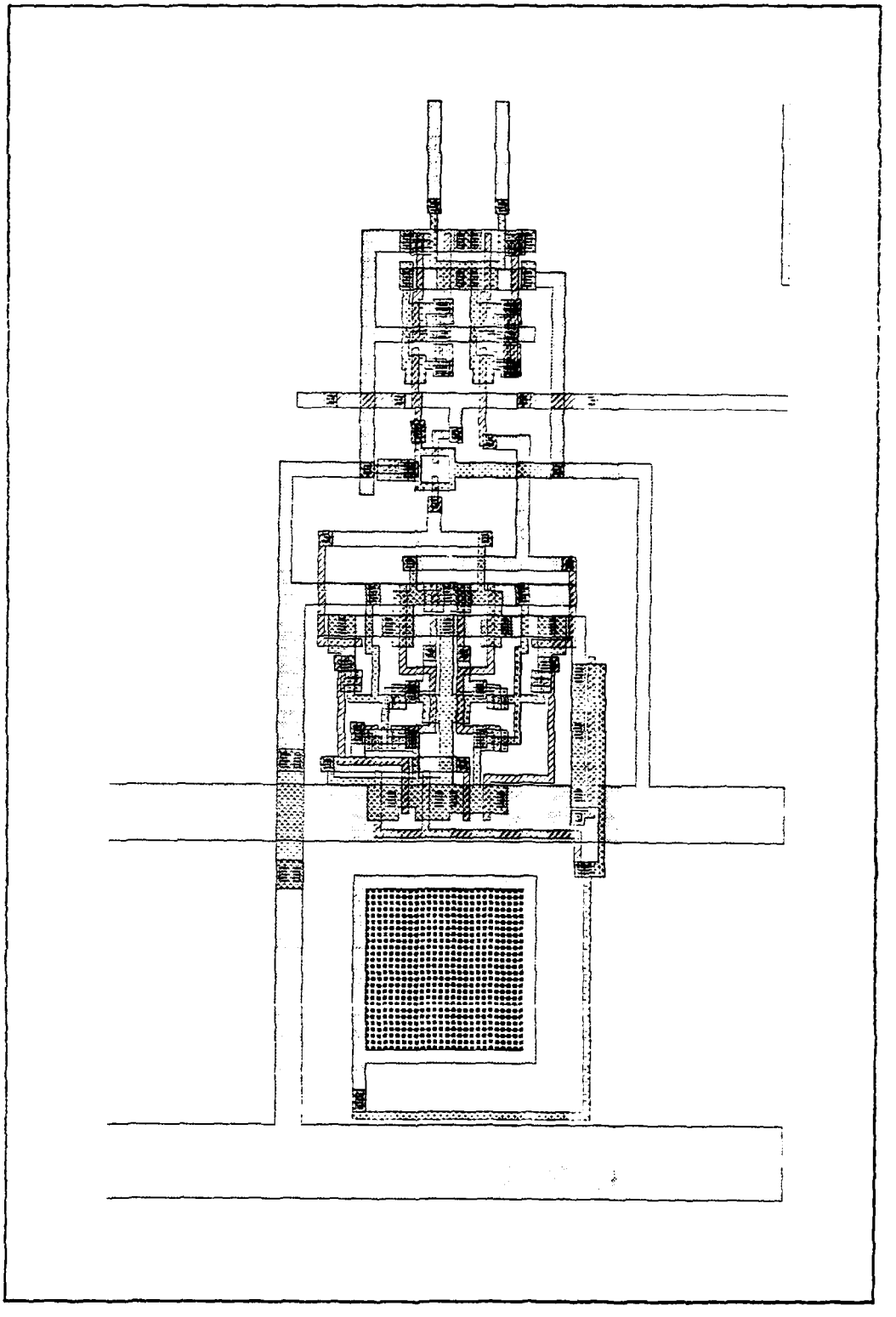


Figure B-4. Clock OR Gate and Buffer Cifplot

ESIMs

```
| T-chain test
I
I
I
I
I
I
h Vdd drdy
| drdy high allows outputs to appear at named nodes
1 GND
V tphi1 0100
V tphi2 0001
w t1 a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 a10
R 64
v
w -t1 -a0 -a1 -a2 -a3 -a4 -a5 -a6 -a7 -a8 -a9 -a10
W tcount a10 a9 a8 a7 a6 a5 a4 a3 a2 a1 a0
| 4(2^^11)-64already done
R 8128
v
| tcount should equal 2^^11=2048 here
h trst
R 10
v
q
| final tcount should be 0 due to trst
839 transistors, 656 nodes (417 pulled up)
initialization took 2012 steps
initialization took 335 steps
initialization took 4 steps
initialization took 0 steps
initialization took 0 steps
initialization took 0 steps
>1110000111100001111000011110000111100001111
                                    00001111000011110000l:t1
>1110000111100001111000011110000111100001111
                                    00001111000011110000l:a0
>1110000000011111111000000001111111100000000
                                    11111111000000001111l:a1
>1110000000000000000111111111111111100000000
                                    00000000111111111111l:a2
>1110000000000000000000000000000000011111111
                                    11111111111111111111:a3
>1110000000000000000000000000000000000000000
                                    00000000000000000000:a4
>1110000000000000000000000000000000000000000
                                    00000000000000000000:a5
>1110000000000000000000000000000000000000000
                                    00000000000000000000:a6
```

```
>11100000000000000000000000000000000000000
                                        00000000000000000000:a7
>11100000000000000000000000000000000000000
                                        00000000000000000000:a8
>11100000000000000000000000000000000000000
                                        00000000000000000000:a9
>11100000000000000000000000000000000000000
                                        00000000000000000000:a10
a10=0 a9=0 a8=0 a7=0 a6=0 a5=0 a4=0 a3=1 a2=1 a1=1 a0=1 t1=1
h inputs: Vdd drdy tphi2
1 inputs: GND tphi1
tcount=11111111111      2047
h inputs: Vdd drdy tphi2
1 inputs: GND tphi1
tcount=00000000000      0
h inputs: Vdd drdy trst tphi1
1 inputs: GND tphi2
839 transistors, 656 nodes (417 pulled up)

| C-chain test
I
I
I
I
I
I
h Vdd
1 GND
V cphi1 0100
V cphi2 0001
w c1 1sb nsb msb c21 c22 c23 c24
W cvalue c24 c23 c22 c21 msb nsb 1sb c1
R 64
| cvalue should be 16 at this point
v
w -c1 -1sb -nsb -msb -c21 -c22 -c23 -c24
h crst
R 10
v
| final cvalue should be zero because crst is high
q
839 transistors, 656 nodes (417 pulled up)
initialization took 2012 steps
initialization took 335 steps
initialization took 4 steps
initialization took 0 steps
initialization took 0 steps
initialization took 0 steps
>11100001111000011110000111100001111000011111
                                        00001111000011110000:c1
>11100000000111111111000000001111111100000000
                                        11111111000000001111:1sb
>11100000000000000001111111111111111100000000
                                        00000000111111111111:nsb
```

```
>11100000000000000000000000000000011111111
                                        11111111111111111111:msb
>11100000000000000000000000000000000000000000
                                        00000000000000000000:c21
>11100000000000000000000000000000000000000000
                                        00000000000000000000:c22
>11100000000000000000000000000000000000000000
                                        00000000000000000000:c23
>11100000000000000000000000000000000000000000
                                        00000000000000000000:c24
cvalue=00001111 15
c24=0 c23=0 c22=0 c21=0 msb=1 nsb=1 lsb=1 c1=1
h inputs: Vdd cphi2
1 inputs: GND cphi1
cvalue=00000000 0
h inputs: Vdd crst cphi1
1 inputs: GND cphi2
839 transistors, 656 nodes (417 pulled up)


| control logic test
I
I
I
I
I
I
h Vdd gate
| run with gate high to enable control logic nand gates
1 GND
V cphi1 0100
V cphi2 0001
w cphi1 cphi2 c1
w gate msb nsb lsb
w drdy trst crst ndrdy ntrst ncrst
R 120
| run through 30 c-osc clock cycles
q
839 transistors, 656 nodes (417 pulled up)
initialization took 2012 steps
initialization took 335 steps
initialization took 4 steps
initialization took 0 steps
initialization took 0 steps
initialization took 0 steps
>01000100010001000100010001000100010001
00010001000100010001000100010001000100010001000
10001000100010001000100010001000100010001000100:cphi1
>00010001000100010001000100010001000
01000100010001000100010001000100010001000100010
00100010001000100010001000100010001000100010001:cphi2
>11100001111000011110000111100001111
10000111111111000011111000011110000111111110O
00111100001111000011111111000011110000111110:c1
```

```
>11111111111111111111111111111111
11111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111:gate
>11100000000000000000000000000000
01111111100000000000000000000011111111000000
00000000000000111111110000000000000000000001:msb
>11100000000000000001111111111111
10000000000001111111111111111100000000000011
11111111111111000000000000111111111111111110:nsb
>11100000000111111110000000011111
10000000011110000000011111111100000000111100
00000011111111000000001111000000011111111110:lsb
>00000000000111111110000000000000
00000000011110000000000000000000000000111100
00000000000000000000001111000000000000000000:drdy
>00000000000000000000000000001111111
10000000000000000000011111111100000000000000
00000111111111000000000000000000001111111110:trst
>00000000000000000000000000000000
00000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000:crst
>11111111111000000001111111111111
11111111100001111111111111111111111111000011
11111111111111111111110000111111111111111111:ndrdy
>11111111111111111111111110000000
01111111111111111111000000001111111111111111
11111000000001111111111111111111111000000001:ntrst
>11111111111111111111111111111111
11111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111:ncrst
839 transistors, 656 nodes (417 pulled up)


| control logic test
I
I
I
I
I
I
h Vdd
1 GND gate
| run with gate low to disable control logic nand gates
V cphi1 0100
V cphi2 0001
w cphi1 cphi2 c1
w gate msb nsb lsb
w drdy trst crst ndrdy ntrst ncrst
R 120
| run through 30 c-osc clock cycles
q
839 transistors, 656 nodes (417 pulled up)
initialization took 2012 steps
initialization took 335 steps
```

```
initialization took 4 steps
initialization took 0 steps
initialization took 0 steps
initialization took 0 steps
>0100010001000100010001000100010001
0001000100010001000100010001000100010001000
1000100010001000100010001000100010001000100:cphi1
>0001000100010001000100010001000100
0100010001000100010001000100010001000100010
0010001000100010001000100010001000100010001:cphi2
>1110000111100001111000011110000111
1000011110000111100001111000011110000111100
0011110000111100001111000011110000111100001:c1
>0000000000000000000000000000000000
0000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000:gate
>1110000000000000000000000000000000
0111111111111111111111111111111110000000000
0000000000000000000001111111111111111111111:msb
>1110000000000000001111111111111111
1000000000000000011111111111111110000000000
0000011111111111111111000000000000000011111:nsb
>1110000000011111111000000001111111
1000000001111111100000000111111110000000011
1111110000000011111111100000000111111100000:lsb
>0000000000000000000000000000000000
0000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000:drdy
>0000000000000000000000000000000000
0000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000:trst
>0000000000000000000000000000000000
0000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000:crst
>1111111111111111111111111111111111
1111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111:ndrdy
>1111111111111111111111111111111111
1111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111:ntrst
>1111111111111111111111111111111111
1111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111:ncrst
839 transistors, 656 nodes (417 pulled up)
```

Sample DRC Output

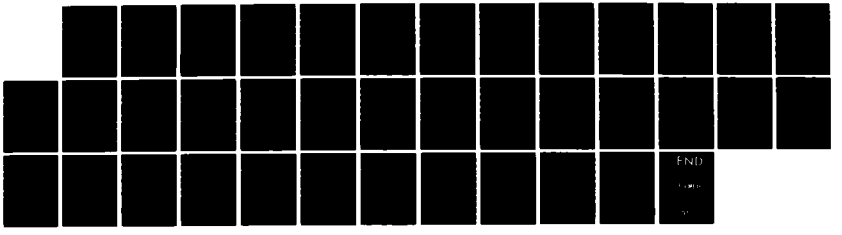
```
Boeing-Modified CMU DRC 1.1 on Wed Jun 27
                19:22:58 EDT 1984 : file dcsol7f.cif
lambda 2.5 microns
----------
WARNING: CONTACT OVERLAP CHECKS DISABLED
WARNING: POLY, DIFF & CUT WIDTH CHECKS DISABLED
----------
begin parsing dcsol7f.cif
total commands parsed: 3395
total storage used 77504 bytes, in 2525 requests
the bounding box is l = 0, r = 862500, b = 0, t = 850000
dcsol7f.NP : 4060 rectangles
dcsol7f.ND : 4232 rectangles
dcsol7f.NI : 427 rectangles
dcsol7f.NM : 3984 rectangles
dcsol7f.NC : 3334 rectangles
dcsol7f.NG : 40 rectangles
dcsol7f.NB : 0 rectangles
Identify butting contacts :
First, coalesce adjacent contact-cut rectangles
dcsol7f.Nc : 3200 rectangles
Second, find contacts which overlap diffusion
dcsol7f.DC : 2711 rectangles
Third, find the subset of those contacts which overlap poly
dcsol7f.BC : 445 rectangles
Form expanded butting contact rectangles
dcsol7f.BCexp : 445 rectangles
Find the edges of the butting contact regions
dcsol7f.BUTC : 1780 edges
Find (poly AND diffusion) rectangles
dcsol7f.PD : 3080 rectangles
Find the gate (poly-diff not butting contact) rectangles
dcsol7f.GT : 2352 rectangles
Find the gate edges
dcsol7f.GATE : 5896 edges
Find the depletion-mode channel rectangles
dcsol7f.DG : 488 rectangles
Find the depletion-mode channel edges
dcsol7f.DGAT : 1700 edges
Find the enhancement-mode channel rectangles
dcsol7f.EG : 1896 rectangles
Find the enhancement-mode channel edges
dcsol7f.EGAT : 4196 edges
Find the contact-in-diffusion rectangles
dcsol7f.cd : 3486 rectangles
Exclude the butting contacts from the diffusion-contacts
```

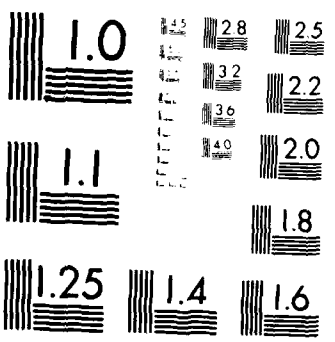
```
dcsol7f.CD : 2940 rectangles
Find the contact-in-diffusion edges
dcsol7f.dCUT : 8824 edges
Find non-transistor diffusion (source and drain) rectangles
dcsol7f.SR : 6834 rectangles
Find non-transistor diffusion (source and drain) edges
dcsol7f.SRCE : 15764 edges
Find diffusion edges
dcsol7f.DIFF : 9892 edges
Check spacing of diffusion
72 violations listed in uerr.dS.dcsol7f
0 violations remain in err.dS.dcsol7f after
                                    filtering spurious messages
Find implant edges
dcsol7f.IMPL : 1668 edges
Check implant overlap of gate
Check implant-gate spacing
Find poly edges
dcsol7f.POLY : 9744 edges
Check spacing of poly
105 violations listed in uerr.pS.dcsol7f
0 violations remain in err.pS.dcsol7f after
                                    filtering spurious messages
Check minimum extension of poly past gate
Check poly-diffusion spacing
300 violations listed in uerr.pSd.dcsol7f
0 violations remain in err.pSd.dcsol7f
                          after filtering spurious messages
Find contact-cut edges
dcsol7f.CUT : 12560 edges
Check spacing of cut
Find metal edges
dcsol7f.METL : 8344 edges
Check diffusion-cut to gate spacing
Check width of metal
Check spacing of metal
1 violations listed in uerr.mS.dcsol7f
0 violations remain in err.mS.dcsol7f
                          after filtering spurious messages

That finishes this run of the DRC.
No design rule violations were found.
You have a beautiful chip!
```

AD-A151 907 DEVELOPMENT OF A DIGITALLY COMPENSATED SAW OSCILLATOR (DCSO) TO PROVIDE T.. (U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. W D COWAN 2/2

UNCLASSIFIED DEC 84 AFIT/GE/ENG/84D-22 F/G 9/5 NL





```
/*************************************************************
*                                                            *
*       DATE:  10/16/84                                      *
*       VERSION:  1.10                                       *
*       NAME:  read_freq                                     *
*       FUNCTION: reads frequency measured by HP 5340A       *
*       INPUTS: tempout-pointer to 16 char buffer, aport,    *
*                                                  bport     *
*       OUTPUTS: fills buffer                                *
*       CALLING MODULES: as needed                           *
*                                                            *
*       AUTHOR: William D. Cowan                             *
*       HISTORY: Modified read_freq version 1.00 to look     *
*       for DAV at bport bit 8 instead of aport bit 1.       *
*       Requires a change to the interface cable but         *
*       allows avoidance of cranky pio port on S-100         *
*       system.                                              *
*                                                            *
*************************************************************/


read_freq(tempout,aport,bport)
char *tempout;
int aport;
int bport;

{
int cntr;
int dav;

dav=1;
for(cntr=0;cntr < 16;cntr=cntr+1)
{
outp(aport,2);  /* RFD=0, DAC=1 -- counter stops */
outp(aport,0);  /* RFD=DAC=0 */
outp(aport,4);  /* RFD=1,DAC=0 -- send me a data byte */
while (dav)
  dav=(inp(bport) & 128);  /* check for DAV
                              - data available */
*(tempout+cntr)=((inp(bport) ^ 255) & 127);
        /* exclusive OR corrects for inverted
                                 data from HP 5340A */
        /* AND removes most significant bit */
}       /* end of FOR loop */
outp(aport,6);  /* restart counter */
return;
}       /* end of read_freq */
```

```
/*************************************************************
*                                                           *
*       DATE:  10/16/84                                     *
*       VERSION:  1.10                                      *
*       NAME:  read_temp                                    *
*       FUNCTION: reads temperature measured by cyborg via  *
*       a hardware port, and puts the reading in a five     *
*       character buffer supplied by the calling routine.   *
*                                                           *
*       INPUTS:  tempout-a pointer to the first character   *
*       of the buffer supplied by the caller                *
*       CALLING MODULES: as needed                          *
*                                                           *
*       AUTHOR:  William D. Cowan                           *
*       HISTORY:  Slight modification of read_temp version  *
*       1.00 to allow the calling routine to provide the    *
*       port number to which the cyborg is connected.       *
*       Change simplifies initialization by allowing all    *
*       ports to be defined in a single section of the      *
*       main program.                                       *
*                                                           *
*************************************************************/

read_temp(tempout,cport)
char *tempout;
int cport;

{
int portin;

portin=inp(cport);
while(!(portin & 128))
        portin=inp(cport);
*tempout=((portin & 15)+0x30);
while(!(portin & 64))
        portin=inp(cport);
*(tempout+1)=((portin & 15)+0x30);
while(!(portin & 32))
        portin=inp(cport);
*(tempout+2)='.';
*(tempout+3)=((portin & 15)+0x30);
while(!(portin & 16))
        portin=inp(cport);
*(tempout+4)=((portin & 15)+0x30);
return;
}
```

```
/*************************************************************
*                                                            *
*       DATE: 10/17/84                                       *
*       VERSION: 1.00                                        *
*       TITLE: DCSO Frequency vs Temperature Test            *
*       FILENAME: dcsotest.c                                 *
*       OWNER: William D. Cowan                              *
*       SOFTWARE SYSTEM: BDSC Ver 1.5                        *
*       OPERATING SYSTEM: CP/M                               *
*       LANGUAGE: C                                          *
*       USE: compile, link and run                           *
*       CONTENTS: main                                       *
*       FUNCTION: reads hardware ports to which HP5340A and  *
*       Cyborg are connected and displays selected data      *
*       points                                               *
*                                                            *
*************************************************************/

#include <a:bdscio.h>

/*************************************************************
*                                                            *
*       DATE:  10/16/84                                      *
*       VERSION: 1.00                                        *
*       NAME: main (for freq vs temp)                        *
*       MODULE NUMBER: n/a                                   *
*       FUNCTION: uses read_freq and read_temp to take       *
*       data for temperature vs frequency plots              *
*       INPUTS: n/a                                          *
*       OUTPUTS: temperature and frequency to crt            *
*       MODULES CALLED: read_freq,read_temp                  *
*                                                            *
*       AUTHOR:  William D. Cowan                            *
*       HISTORY:                                             *
*                                                            *
*************************************************************/

main()
{
int aport,bport,cport,notdone;
char freqbuf[16];
char tempbuf[5];

/* assign port numbers */
aport=247;      /* out */
bport=244;      /* in */
cport=246;      /* in */

/* initialize variables */
notdone=1;      /* notdone true */

/* data taking loop */
while (notdone)
```

```
    {
    read_temp(&tempbuf[0],cport);
/* if statement controls the data points displayed
                        - in this case X2.0X degrees C  */
    if ((*(tempbuf+1)%2==0) && (*(tempbuf+3)==0x30))
        {
        read_freq(&freqbuf[0],aport,bport);
        printf("                ");          /*margin*/
        printf("%4s",tempbuf);
        printf("     ");
        printf("%-14.12s\n",(freqbuf+2));
        }       /* endif */
    }   /* endwhile */
}       /* end main */
```

```
/*************************************************************
*                                                            *
*       DATE: 10/17/84                                       *
*       VERSION: 1.00                                        *
*       TITLE: Manual DCSO Frequency vs Temperature Test     *
*       FILENAME: takedata.c                                 *
*       OWNER: William D. Cowan                              *
*       SOFTWARE SYSTEM: BDSC Ver 1.5                        *
*       OPERATING SYSTEM: CP/M                               *
*       LANGUAGE: C                                          *
*       USE: compile, link and run                           *
*       CONTENTS: main                                       *
*       FUNCTION: reads hardware ports to which HP5340A      *
*       and Cyborg are connected and displays manually       *
*       selected data points                                 *
*                                                            *
*************************************************************/

#include <a:bdscio.h>

/*************************************************************
*                                                            *
*       DATE:  10/16/84                                      *
*       VERSION: 1.00                                        *
*       NAME: main (for takedata)                            *
*       MODULE NUMBER: n/a                                   *
*       FUNCTION: uses read_freq and read_temp to take       *
*       data for temperature vs frequency plots              *
*       INPUTS: n/a                                          *
*       OUTPUTS: temperature and frequency to crt            *
*       MODULES CALLED: read_freq,read_temp                  *
*                                                            *
*       AUTHOR:  William D. Cowan                            *
*       HISTORY:                                             *
*                                                            *
*************************************************************/

main()
{
int aport,bport,cport,notdone,key;
char freqbuf[16];
char tempbuf[5];

/* assign port numbers */
aport=247;      /* out */
bport=244;      /* in */
cport=246;      /* in */

/* initialize variables */
notdone=1;      /* notdone true */
key=0;          /* false */

/* data taking loop */
```

```
while (notdone)
     {
     while (!key)
          {
          key=kbhit();
          }
     read_temp(&tempbuf[0],cport);
     read_freq(&freqbuf[0],aport,bport);
     printf("                ");     /*margin*/
     printf("%4s",tempbuf);
     printf("      ");
     printf("%-14.12s\n",(freqbuf+2));
     key=0;        /* reset to false */
     }    /* endwhile */
}         /* end main */
```

```
/*************************************************************
*                                                            *
*       DATE:  8/24/84                                       *
*       VERSION:  1.00                                       *
*                                                            *
*       TITLE:  Chiptest                                     *
*       FILENAME: chiptest.c                                 *
*       OWNER:  William D. Cowan                             *
*       SOFTWARE SYSTEM:  BDS C                              *
*       USE: compile, link, run                              *
*       CONTENTS: chip test driver and called modules        *
*       FUNCTION:  automate testing of DCSO chip             *
*                                                            *
*************************************************************/

#include <a:bdscio.h>

/*************************************************************
*                                                            *
*       DATE:  8/24/84                                       *
*       VERSION:  1.00                                       *
*                                                            *
*       NAME:  main                                          *
*       MODULES CALLED: cchn_rst, cchn_test, tchn_test       *
*                       logic_tst, (strbbits)                *
*       AUTHOR:  William D. Cowan                            *
*       HISTORY:                                             *
*                                                            *
*************************************************************/

main()

{
int choice,acon,adat,bcon,bdat,cdat,bitport;
int abitmsk,inport,done;

/* initializing ports */
acon=6;
adat=4;
bcon=7;
bdat=5;
cdat=245;
bitport=0xcf;
abitmsk=0xfe;
inport=0x4f;

/* port setup */
outp(acon,bitport);      /* note: cdat port is set */
outp(acon,abitmsk);      /* for input via hardware */
outp(bcon,inport);

/* menu loop */
while (!done)
```

```
    {
    printf("                              William Cowan \n");
    printf("                                 8-24-84 \n");
    printf("\n");
    printf("\n");
    printf("                DCSO Version 1.00 - Chip Test \n");
    printf("\n");
    printf("     1.  C-Chain Reset \n");
    printf("     2.  C-Chain Test \n");
    printf("     3.  Control Logic Test \n");
    printf("     4.  T-Chain Test \n");
    printf("     5.  Complete Test \n");
    printf(" \n");
    printf("Test Selection (control-c to abort)? ");
    choice=0;
    while (!choice)
        {
        choice=getchar();
        if (choice < 48 || choice > 53)
            {
            printf("\n Invalid Selection - Try Again \n");
            choice=0;
            }   /* endif */
        }       /* endwhile */
    if (choice == 49)
        cchn_rst(adat,bdat);
    if (choice == 50)
        cchn_test(adat,bdat);
    if (choice == 51)
        logic_tst(adat);
    if (choice == 52)
        tchn_test(adat,bdat,cdat);
    if (choice == 53)
        {
        logic_tst(adat);
        tchn_test(adat,bdat,cdat);
        cchn_rst(adat,bdat);
        cchn_test(adat,bdat);
        }       /* endif */
    }   /* endwhile */
}       /* end main */


/*************************************************************
*                                                            *
*       DATE:  8/24/84                                       *
*       VERSION:  1.00                                       *
*                                                            *
*       NAME:  cchn_rst (C-Chain reset)                      *
*       MODULES CALLED: strbbits                             *
*       CALLING MODULES: main of chiptest.c                  *
*                                                            *
*       AUTHOR:  William D. Cowan                            *
*       HISTORY:                                             *
```

```
*                                                                    *
**********************************************************************/

cchn_rst(adat,bdat)

int adat,bdat;
{
int errchk,crst,quit;

errchk=0;
crst=0;
printf("\n");
printf("C-Chain Reset \n");
printf("============= \n");
outp(adat,0);    /* set gatein low */
/* stobe T-Osc. until upper bits are high */
while (!((inp(bdat) & 14) == 14))
    strbbits(adat,1);
outp(adat,4);    /* set gate high to generate reset */
quit=0;
/* strobe C-Osc until high bits all go low showing reset */
while ((!((inp(bdat) & 14) == 0)) | quit)
    {
    strbbits(adat,1);
    errchk = errchk + 1;
    if (errchk >= 20)    /* should take < 15 counts */
        {
        printf("\n ERROR -- No C-Reset \n");
        quit=1;          /* TRUE */
        }        /* endif */
    }    /* endwhile */
if (quit == 0)
    printf("\n C-Chain Reset \n");
return;
}        /* end cchn_rst */


/**********************************************************************
*                                                                     *
*       DATE:  8/24/84                                                *
*       VERSION:  1.00                                                *
*                                                                     *
*       NAME:  cchn_test (C-Chain test)                               *
*       MODULES CALLED: strbbits                                      *
*       CALLING MODULES: main of chiptest.c                           *
*                                                                     *
*       AUTHOR:  William D. Cowan                                     *
*       HISTORY:                                                      *
*                                                                     *
**********************************************************************/

cchn_test(adat,bdat)

int adat,bdat;
```

```
{
int c_out,cntr1,cntr2,done,oldc_out;

/* initializing variables */
cntr1=0;
cntr2=0;
done=0;
oldc_out=0;

outp(adat,0);   /* set gatein low */

/* print output heading */
printf("\n");
printf("C-Chain Test \n");
printf("============ \n");
printf("  C21 C22 C23 C24      Count \n");
printf("  --- --- --- ---      ----- \n");

/* test loop */
while (!done)
    {
    if (cntr1 == 10000)
        {
        cntr2 = cntr2 + 1;
        cntr1 = 0;
        }       /* endif */
    strbbits(adat,1);   /* strobe C-Osc. */
    c_out = (inp(bdat) & 15);
    cntr1 = cntr1 + 1;
    if (c_out != oldc_out)
        {
        printf("%4d",(c_out & 1));
        printf("%4d",((c_out & 2)/2));
        printf("%4d",((c_out & 4)/4));
        printf("%4d",((c_out & 8)/8));
        if (cntr2 != 0)
            printf("%9d",cntr2);
        printf("%04d",cntr1);
        printf("\n");
        }       /* endif */
    oldc_out = c_out;
    done = (inp(adat) & 64);    /* check for C-Reset */
    if (cntr2 >= 1680)
        {
        printf("ERROR -- C-Chain count overflow \n");
        done=1; /* true */
        }       /* endif */
    }   /* endwhile - test loop */
return;
}       /* end cchn_test */


/*****************************************************************
```

```
*                                                                    *
*       DATE:  8/24/84                                               *
*       VERSION:  1.00                                               *
*                                                                    *
*       NAME:  tchn_test (T-Chain test)                              *
*       MODULES CALLED: strbbits                                     *
*       CALLING MODULES: main of chiptest.c                          *
*                                                                    *
*       AUTHOR:  William D. Cowan                                    *
*       HISTORY:                                                     *
*                                                                    *
**********************************************************************/

tchn_test(adat,bdat,cdat)

int adat,bdat,cdat;

{
int t_out,errchk,datardy,cntr,done,strb;

outp(adat,4);   /* set gatein high */

/* set datardy high to enable latches */
errchk=0;
while (datardy)          /* actually looking for ~DataRdy */
    {
    strbbits(adat,1);
    datardy = (inp(adat) & 8);
    errchk = errchk + 1;
    if (errchk >= 33)
        {
        printf("ERROR -- No ~DataRdy for T-Chain Test \n");
        exit();
        }       /* endif */
    }   /* endwhile */

/* print output heading */
printf("\n");
printf("T-Chain Test \n");
printf("============ \n");
printf("A0 - - - - - - - - - - A11     Count \n");
printf("---------------------------     ----- \n");

/* initializing for main test loop */
cntr=0;
done=0;
errchk=0;
strb=1;

/* main test loop */
while (!done)
    {
    t_out = ((inp(cdat) * 16) + (inp(bdat)/16));
    /* print output */
```

```
        printf("%2d",(t_out & 1));
        printf("%2d",((t_out & 2)/2));
        printf("%2d",((t_out & 4)/4));
        printf("%2d",((t_out & 8)/8));
        printf("%2d",((t_out & 16)/16));
        printf("%2d",((t_out & 32)/32));
        printf("%2d",((t_out & 64)/64));
        printf("%2d",((t_out & 128)/128));
        printf("%2d",((t_out & 256)/256));
        printf("%2d",((t_out & 512)/512));
        printf("%2d",((t_out & 1024)/1024));
        printf("%2d",((t_out & 2048)/2048));
        printf("%10d \n",cntr);

    /* to provide varying printing intervals */
        if (cntr < 15)
            strb=1;
        if (cntr >= 15)
            strb=16;
        if (cntr >= 255)
            strb=256;
        while (strb)          /* strb not equal 0 */
            {
            strbbits(adat,2);
            cntr = cntr + 1;
            strb = strb - 1;
            }          /* endwhile */
        if (cntr >= 4100)    /* max count 2^12=4096 */
            done=1; /* true */
        if (errchk > 5000)
            {
            printf("ERROR -- T-Chain Test Timeout \n');
            exit();
            }          /* endif */
        }    /* endwhile - main test loop */
    return;
    }        /* end tchn_test */


    /*************************************************************
    *                                                            *
    *       DATE:  8/24/84                                       *
    *       VERSION:  1.00                                       *
    *                                                            *
    *       NAME:  logic_tst (Control logic test)                *
    *       MODULES CALLED:                                      *
    *       CALLING MODULES: main of chiptest.c                  *
    *                                                            *
    *       AUTHOR:  William D. Cowan                            *
    *       HISTORY:                                             *
    *                                                            *
    *************************************************************/

    logic_tst(adat)
```

```
int adat;

{
int cntr,testword,errchk,crst;

outp(adat,4);   /* set gatein high */
errchk=0;
while (! crst)
    {
    strbbits(adat,1);
    crst = (inp(adat) & 64);
    errchk = errchk + 1;
    if (errchk > 20)
        {
        printf("ERROR -- No C-Reset in C L Test \n");
        exit();
        }       /* endif */
    }   /* endwhile */

/* print output heading */
printf("\n");
printf("Control Logic Test \n");
printf("================== \n");
printf("  ~DR  TR ~TR  CR ~CR    Count \n");
printf("  --- --- --- --- ---    ----- \n");

/* initialize counter */
cntr=0;

/* main test loop */
while (cntr <= 16)
    {
    strbbits(adat,1);
    testword=inp(adat);
    printf("%4d",((testword & 8)/8));
    printf("%4d",((testword & 16)/16));
    printf("%4d",((testword & 32)/32));
    printf("%4d",((testword & 64)/64));
    printf("%4d",((testword & 128)/128));
    printf("%8d \n",cntr);
    cntr=cntr+1;
    }   /* endwhile - test loop */
return;
}       /* end logic_tst */


/*************************************************************
*                                                            *
*       DATE:  8/24/84                                       *
*       VERSION:  1.00                                       *
*                                                            *
*       NAME:  strbbits (Strobe bits)                        *
*       CALLING MODULES: cchn_rst, cchn_test, tchn_test,     *
```

```
*                          logic_test                                  *
*       AUTHOR:  William D. Cowan                                      *
*       HISTORY:                                                       *
*                                                                      *
**********************************************************************/

strbbits(port,bits)

int port,bits;

{
int high,low;

high=(inp(port) | bits);
outp(port,high);
low=(high & (!bits));
outp(port,low);
return;
}       /* end strbbits */
```

```
/*************************************************************
*                                                            *
*       DATE:  8/27/84                                       *
*       VERSION:  1.00                                       *
*                                                            *
*       TITLE:  Watch                                        *
*       FILENAME: watch.c                                    *
*       OWNER:  William D. Cowan                             *
*       SOFTWARE SYSTEM:  BDS C                              *
*       USE: compile, link, run                              *
*       CONTENTS: self contained                             *
*       FUNCTION:  monitors outputs of DCSO chip for use     *
*       when testing with external clocking etc.             *
*                                                            *
*************************************************************/

#include <a:bdscio.h>

/*||    */
/*************************************************************
*                                                            *
*       DATE:  8/27/84                                       *
*       VERSION:  1.00                                       *
*                                                            *
*       NAME:  main                                          *
*       MODULES CALLED:                                      *
*       AUTHOR:  William D. Cowan                            *
*       HISTORY:                                             *
*                                                            *
*************************************************************/

main()

{
int adat,bdat,cdat;

adat=247;
bdat=244;
cdat=245;

watch(adat,bdat,cdat);
}       /* end main */


watch(adat,bdat,cdat)

int adat,bdat,cdat;

{
int a,b,c,old_a,old_b,old_c,cntr1,cntr2,done;

done=0;
while (!done)
```

```
        {
        cntr1=0;
        cntr2=0;

        while (!((a != old_a) || (b != old_b) || (c != old_c)))
            {
            a=(inp(adat) & 0xfc);
            b=inp(bdat);
            c=inp(cdat);
            cntr1=cntr1+1;
            if (cntr1 == 10000)
                {
                cntr2=cntr2+1;
                cntr1=0;
                }   /* endif */
            }       /* end while */

        printf("%1d",((a & 8)/8));          /* ~DataRdy */
        printf("%3d",((a & 16)/16));        /* T-Reset */
        printf("%3d",((a & 32)/32));        /* ~T-Reset */
        printf("%3d",((a & 64)/64));        /* C-Reset */
        printf("%3d",((a & 128)/128));      /* ~C-Reset */
        printf("%6d",(b & 1));              /* C21 */
        printf("%3d",((b & 2)/2));          /* C22 */
        printf("%3d",((b & 4)/4));          /* C23 */
        printf("%3d",((b & 8)/8));          /* C24 */
        printf("%6d",((b & 16)/16));        /* A0 */
        printf("%3d",((b & 32)/32));        /* A1 */
        printf("%3d",((b & 64)/64));        /* A2 */
        printf("%3d",((b & 128)/128));      /* A3 */
        printf("%3d",(c & 1));              /* A4 */
        printf("%3d",((c & 2)/2));          /* A5 */
        printf("%3d",((c & 4)/4));          /* A6 */
        printf("%3d",((c & 8)/8));          /* A7 */
        printf("%3d",((c & 16)/16));        /* A8 */
        printf("%3d",((c & 32)/32));        /* A9 */
        printf("%3d",((c & 64)/64));        /* A10 */
        printf("%3d",((c & 128)/128));      /* A11 */
        if (cntr2 != 0)
            printf("%6d",cntr2);
        else
            printf("      ");
        printf("%04d \n",cntr1);

        old_a = a;
        old_b = b;
        old_c = c;
        }   /* endwhile */
    }       /* end watch */
```

```
10 '*********************************************************
15 '*                                                       *
20 '*                      DCSOCAL.BAS                      *
25 '*                                                       *
30 '*   WILLIAM D. COWAN                          10/10/84  *
35 '*                                                       *
40 '*-------------------------------------------------------*
45 '*  DESCRIPTION:  DCSOCAL serves as a calibration        *
50 '*  program for the AFIT DCSO.  It monitors the          *
55 '*  clock loop frequency via the HP5340A interface       *
60 '*  and based on the reading adjusts the phase           *
65 '*  shifter control word to hold the frequency at a      *
70 '*  stable value as determined by a data statement.      *
75 '*  The frequency read, deviation from desired freq,     *
80 '*  phase shift word (in hex and dec), and T-Chain       *
85 '*  count (in hex and dec) are all displayed for         *
90 '*  data to be taken manually.                           *
95 '*********************************************************
100 ACONT=6
110 ADATA=4
120 BCONT=7
130 BDATA=5
140 BINPORT=&H4F
150 ABITPORT=&HCF
160 AMASK=&HF9
170 OUT ACONT,ABITPORT
180 OUT ACONT,AMASK
190 DIM F(16)    'STORAGE FOR FREQUENCY READINGS
200 DIM D(8)     'CONSTANT=DESIRED FREQUENCY STORAGE
210 FOR I=1 TO 8:READ D(I):NEXT I
220 SHFTWD=3584 'E00HEX FOR STARTERS
230 OUT ADATA,6 :RFD=DAC=1
240 OUT BCONT,BINPORT
250 FOR I=1 TO 16
260 OUT ADATA,2 'RFD=0, DAC=1
270 OUT ADATA,0 'RFD=DAC=0
280 OUT ADATA,4 'RFD=1, DAC=0
290 DAVCHEK=0
300 DAV=(INP(ADATA) AND 1)
310 DAVCHEK=DAVCHEK+1
320 'IF DAVCHEK>10 THEN PRINT "ERROR-NO DAV":END
330 IF DAV=1 THEN GOTO 300:DAVCHEK=0
340 F(I)=VAL(CHR$(INP(BDATA) XOR 255))
350 OUT ADATA,0 :RFD=DAC=0
360 NEXT I
370 OUT ADATA,6 'RESTART COUNTER
380 IF F(4)<>2 GOTO 250
390 IF F(5)<>9 GOTO 250 'CHECK FOR GOOD READING
400 FOR I=4 TO 12:PRINT USING "#";F(I);:NEXT I
410 PRINT "          ";
420 MULT=0
430 FOR I=1 TO 8
440 IF F(I+3) < D(I) THEN MULT=-1:I=8:GOTO 480
```

```
450 IF F(I+3) > D(I) THEN MULT=1:I=8:GOTO 540
460 NEXT I
470 IF MULT=0 GOTO 250
480 FOR J=8 TO 1 STEP-1
490 T(J)=D(J)-F(J+3)
500 IF T(J)=>0 GOTO 520
510 D(J)=D(J)+10:D(J-1)=D(J-1)-1:GOTO 490
520 NEXT J
530 GOTO 590
540 FOR J=8 TO 1 STEP-1
550 T(J)=F(J+3)-D(J)
560 IF T(J)=>0 GOTO 580
570 F(J+3)=F(J+3)+10:F(J+2)=F(J+2)-1:GOTO 550
580 NEXT J
590 TWEAK=0
600 FOR I=1 TO 8          'CONVERT TO INTEGER
610 TWEAK=TWEAK+(T(I)*(10^(8-I)))
620 NEXT I        'IF INTEGER TOO BIG -- CRASH
630 PRINT TWEAK;
640 PRINT "           ";
650 IF ABS(TWEAK)<10 THEN PRINT "";:GOTO 690
660 SHFTWD=SHFTWD+MULT
670 OUT 244,(SHFTWD\256)
680 OUT 247,(SHFTWD MOD 256)
690 PRINT HEX$(SHFTWD);,SHFTWD;
700 COUNT=(INP(245)\2)+((INP(246) AND 31)*128)
710 PRINT COUNT;,HEX$(COUNT)
720 GOTO 250
730 DATA 2,9,9,0,0,0,0,0
```

```
100 '*********************************************************
110 '*                                                       *
120 '*                      CALCURVE.BAS                     *
130 '*                                                       *
140 '*  WILLIAM D. COWAN                          10/14/84   *
150 '*                                                       *
160 '*-------------------------------------------------------*
170 '*  DESCRIPTION:  CALCURVE is a somewhat kludgey         *
180 '*  program that fits a quadratic equation to the        *
190 '*  calibration data and creates two files called        *
200 '*  LOPROM.MAC and HIPROM.MAC which can be assembled     *
210 '*  and used to program EPROMS.  The least squares       *
220 '*  method is used.  Included but commented out is       *
230 '*  test data from Advanced Engineering Mathematics      *
240 '*  C. Ray Wylie (p. 156).  The example on that page     *
250 '*  was used to develop this program. Also output is     *
260 '*  hard copy of the normal equation coefficients and    *
270 '*  final equation coefficients as well as several       *
280 '*  test points for graphing the calibration curve.      *
290 '*********************************************************
300 DEFDBL A-H,O-Z
310 N=20
320 DIM E(4,4,N)
330 DIM X(N),Y(N)
340 PRINT "Reading data and writing equations"
350 FOR I=1 TO N
360 READ X(I): READ Y(I)
370 E(1,1,I)=1!
380 E(1,2,I)=X(I)
390 E(1,3,I)=(X(I)^2)
400 E(1,4,I)=Y(I)
410 E(2,1,I)=X(I)
420 E(2,2,I)=(X(I)^2)
430 E(2,3,I)=(X(I)^3)
440 E(2,4,I)=(Y(I)*X(I))
450 E(3,1,I)=(X(I)^2)
460 E(3,2,I)=(X(I)^3)
470 E(3,3,I)=(X(I)^4)
480 E(3,4,I)=(X(I)^2*Y(I))
490 NEXT I
500 'LPRINT "Equations"
510 'LPRINT "========="
520 'LPRINT
530 'FOR I=1 TO 3        'PRINT OUT EQUATIONS
540 'FOR K=1 TO N
550 'LPRINT E(I,1,K),E(I,2,K),E(I,3,K),E(I,4,K)
560 'NEXT K
570 'NEXT I
580 '     CALCULATE SUMS AND STORE IN E(4,1-4,1-3)
590 PRINT "Calculating normal equations"
600 FOR I=1 TO 3
610 FOR J=1 TO 4
620 FOR K=1 TO N
```

```
630 E(4,J,I)=E(4,J,I)+E(I,J,K)
640 NEXT K
650 NEXT J
660 NEXT I
670 LPRINT "Normal equations (a+bx+cxx=y)"
680 LPRINT "============================="
690 LPRINT
700 FOR I=1 TO 3
710 LPRINT
720 FOR J=1 TO 4
730 LPRINT E(4,J,I);
740 NEXT J
750 LPRINT
760 NEXT I
770 PRINT "Solving for coefficients of equation"
780 D1=E(4,1,1)*E(4,2,2)*E(4,3,3)
790 D2=E(4,2,1)*E(4,3,2)*E(4,1,3)
800 D3=E(4,3,1)*E(4,1,2)*E(4,2,3)
810 D4=E(4,1,3)*E(4,2,2)*E(4,3,1)
820 D5=E(4,2,3)*E(4,3,2)*E(4,1,1)
830 D6=E(4,3,3)*E(4,1,2)*E(4,2,1)
840 D=D1+D2+D3-D4-D5-D6
850 A1=E(4,4,1)*E(4,2,2)*E(4,3,3)
860 A2=E(4,2,1)*E(4,3,2)*E(4,4,3)
870 A3=E(4,3,1)*E(4,4,2)*E(4,2,3)
880 A4=E(4,4,3)*E(4,2,2)*E(4,3,1)
890 A5=E(4,2,3)*E(4,3,2)*E(4,4,1)
900 A6=E(4,3,3)*E(4,4,2)*E(4,2,1)
910 A=A1+A2+A3-A4-A5-A6
920 B1=E(4,1,1)*E(4,4,2)*E(4,3,3)
930 B2=E(4,4,1)*E(4,3,2)*E(4,1,3)
940 B3=E(4,3,1)*E(4,1,2)*E(4,4,3)
950 B4=E(4,1,3)*E(4,4,2)*E(4,3,1)
960 B5=E(4,4,3)*E(4,3,2)*E(4,1,1)
970 B6=E(4,3,3)*E(4,1,2)*E(4,4,1)
980 B=B1+B2+B3-B4-B5-B6
990 C1=E(4,1,1)*E(4,2,2)*E(4,4,3)
1000 C2=E(4,2,1)*E(4,4,2)*E(4,1,3)
1010 C3=E(4,4,1)*E(4,1,2)*E(4,2,3)
1020 C4=E(4,1,3)*E(4,2,2)*E(4,4,1)
1030 C5=E(4,2,3)*E(4,4,2)*E(4,1,1)
1040 C6=E(4,4,3)*E(4,1,2)*E(4,2,1)
1050 C=C1+C2+C3-C4-C5-C6
1060 A=A/D:B=B/D:C=C/D
1070 LPRINT
1080 LPRINT "Coefficients of equation"
1090 LPRINT "========================="
1100 LPRINT
1110 LPRINT "A= ",A
1120 LPRINT "B= ",B
1130 LPRINT "C= ",C
1140 LPRINT:LPRINT:LPRINT
1150 PRINT "Running test points"
1160 LPRINT "Test points (x y)"
```

```
1170 LPRINT "================"
1180 LPRINT
1190 FOR I=1050 TO 1400 STEP 50
1200 LPRINT I;
1210 YTEST=(I^2)*C+I*B+A
1220 LPRINT YTEST
1230 NEXT I
1240 S$="       ;"
1250 CR$=CHR$(13)       'CARRIAGE RETURN
1260 DB$="      DB      "
1270 PRINT "Writing to LOPROM.MAC"
1280 OPEN "O",#1,"LOPROM.MAC"
1290 PRINT#1,";LOPROM.MAC - from calcurve.bas 10/14/84";CR$
1300 PRINT#1,";assemble to provide EPROM file";CR$
1310 PRINT#1,";         Phase Shifter Word   -  Address";CR$
1320 PSLOW=&HFD
1330 FOR I=0 TO 1049    'OUT OF CAL CURVE RANGE
1340 PRINT#1,DB$;PSLOW;S$;I;CR$
1350 NEXT I
1360 FOR I=1050 TO 1400 'USE CAL CURVE DATA
1370 PSLOW=((I^2)*C+I*B+A) MOD 256
1380 PRINT#1,DB$;PSLOW;S$;I;CR$
1390 NEXT I
1400 FOR I=1401 TO 2047 'OUT OF CAL CURVE RANGE
1410 PRINT#1,DB$;PSLOW;S$;I;CR$
1420 NEXT I
1430 PRINT#1,"  END";CR$
1440 CLOSE #1   'CLOSE LOPROM.MAC
1450 PRINT "Writing to HIPROM.MAC"
1460 OPEN "O",#1,"HIPROM.MAC"
1470 PRINT#1,";HIPROM.MAC - from calcurve.bas 10/14/84";CR$
1480 PRINT#1,";assemble to provide EPROM file";CR$
1490 PRINT#1,";         Phase Shifter Word   -  Address";CR$
1500 PSHI=&HE
1510 FOR I=0 TO 1049    'OUT OF CAL CURVE RANGE
1520 PRINT#1,DB$;PSHI;S$;I;CR$
1530 NEXT I
1540 FOR I=1050 TO 1400 'USE CAL CURVE DATA
1550 PSHI=((I^2)*C+I*B+A)\256
1560 PRINT#1,DB$;PSHI;S$;I;CR$
1570 NEXT I
1580 FOR I=1401 TO 2047 'OUT OF CAL CURVE RANGE
1590 PRINT#1,DB$;PSHI;S$;I;CR$
1600 NEXT I
1610 PRINT#1,"  END";CR$
1620 CLOSE #1   'CLOSE HIPROM.MAC
1630 PRINT CHR$(7)      'BELL
1640 PRINT "Done"
1650 END
1660 '  DCSO DATA
1670 DATA 1263,3672,1251,3676,1247,3675,1166,3736,1153,3746
1680 DATA 1124,3774,1110,3773,1095,3795,1073,3809,1066,3815
1690 DATA 1148,3746,1277,3661,1316,3659,1322,3656,1331,3652
1700 DATA 1334,3651,1352,3644,1359,3643,1361,3645,1382,3642
```

Appendix D: AFIT IC Construction Procedure

1. Inspect thick film solder pads for good tinning

   - discard if not well tinned.

2. Silver paint a conductor from one side of the ground conductor bonding area to the center of the die mounting area.

3. Inspect silver painted conductor to insure continuity

   - use an ohmmeter if visually unsure.

4. Using a very small drop of silver paint, bond NMOS controller die to thick film, insuring that die is positioned correctly. Inspect. If silver paint shorts bonding pads, remove die and try again. Excess silver paint may be removed with acetone.

5. Allow silver paint to dry thoroughly, using a warm (about 75 °C) oven to speed drying if necessary.

6. Wire bond connections from die to thick film.

7. Lightly tin leads of two 54S196 TTL prescalers (in 20 pin LCC packages) using 60/40 rosin core solder and low wattage soldering iron.

8. Preheat a hot plate to about 250 °C.

9. Position prescalers on thick film insuring proper orientation.

10. Using tweezers, gently place thick film on hot plate

    - do not allow prescalers to move out of position.

11. Working quickly but gently, press prescalers down onto solder pads as soon as solder flows. Damage to the thick film will result if it is over heated, or heated for too long.

12. Remove thick film from hot plate with tweezers very carefully to avoid disturbing position of prescalers. Allow to cool slowly.

13. Inspect all solder joints with microscope (times 3 magnification) too insure good electrical and mechanical connections.

14. With a low wattage soldering iron and 60/40 solder, solder Berg dip clips to solder pads (top and bottom) - using as little heat as necessary.

15. Remove unsightly excess rosin (from solder) by carefully rinsing thick film with acetetone.

16. Inspect Berg clip solder joints, and wire bonds microscope under.

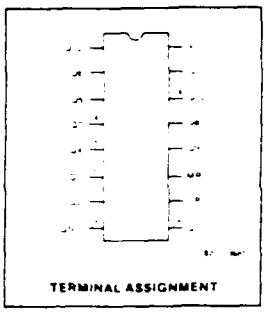
17. Test.

Appendix E: Component Specifications

Component specifications for the counter chips used in the breadboarded control circuitry are provided here because the CD54/74HC4040 parts are relatively new, and have limited availability at the present time. Specificatons for the CD4040 part are provided for comparison. Data sheets for the parts were obtained from the "RCA QMOS High Speed CMOS Logic ICs" (13: 106-110) and "RCA CMOS Integrated Circuits" (12: 122-125) databooks.

Preliminary Data

File Number **1483**



TERMINAL ASSIGNMENT

12-Stage Binary Counter

Type Features:

- *Full static operation*
- *Buffered inputs*
- *Common reset*
- *Negative edge clocking*
- *Typical F_{MAX}=50 MHz @ V_{CC}=5 V, C_L=15 pF*

Family Features:

- *Fanout (Over Temperature Range)*
  - *Standard Outputs - 10 LSTTL Loads*
  - *Bus Driver Outputs - 15 LSTTL Loads*
- *Wide Operating Temperature Range*
  - *CD74HC, HCT, HCU: −40 to +85°C*
- *Balanced Propagation and Transition Times*
- *Significant Power Reduction Compared to LSTTL Logic ICs*
- *Alternate Source is Philips/Signetics*
- *CD54HC, CD74HC Types*
  - *2 to 6 V Operation*
  - *High Noise Immunity: N_{IL}=20%, N_{IH}=30% of V_{CC} @ V_{CC}=5 V*
- *CD54HCT, CD74HCT Types*
  - *4.5 to 5.5 V Operation*
  - *Direct LSTTL Input Logic Compatibility*
    - *V_{IL}=0.8 V Max, V_{IH}=2 V Min*
  - *CMOS Input Compatibility*
    - *$I_{IN} \leq 1\ \mu A$ @ V_{OL}, V_{OH}*

The RCA-CD54/74HC4040 and CD54/74HCT4040 are 12-stage ripple-carry binary counters. All counter stages are master-slave flip-flops. The state of the stage advances one count on the negative transition of each clock pulse a high voltage level on the MR line resets all stages to their zero state. All inputs and outputs are buffered.

The CD54HC4040 and CD54HCT4040 are supplied in 16-lead hermetic dual-in-line ceramic packages (F suffix). The CD74HC4040 and CD74HCT4040 are supplied in 16-lead dual-in-line plastic packages (E suffix).

MAXIMUM RATINGS, *Absolute-Maximum Values*

DC SUPPLY VOLTAGE (V_{CC}):	
Voltages referenced to ground	-0.5 to +7 V
DC INPUT DIODE CURRENT, I_{IK} (FOR $V_I < -0.5$ V OR $V_I > V_{CC}$ +0.5 V)	±20 mA
DC OUTPUT CURRENT, I_{OK} FOR ($V_O < -0.5$ V OR $V_O > V_{CC}$ +0.5 V)	±20 mA
DC DRAIN CURRENT, PER OUTPUT (I_O) (FOR -0.5 V < V_O < V_{CC} + 0.5 V)	±25 mA
DC V_{CC} OR GROUND CURRENT, PER PIN (I_{CC})	±50 mA
POWER DISSIPATION PER PACKAGE (P_D):	
For T_A = -40 to +60°C (PACKAGE TYPE E)	500 mW
For T_A = +60 to +85°C (PACKAGE TYPE E)	Derate Linearly at 8 mW/°C to 300 mW
For T_A = -55 to +100°C (PACKAGE TYPE F)	500 mW
For T_A = +100 to +125°C (PACKAGE TYPE F)	Derate Linearly at 8 mW/°C to 300 mW
OPERATING TEMPERATURE RANGE (T_A):	
PACKAGE TYPE F	-55 to +125°C
PACKAGE TYPE E	-40 to +85°C
STORAGE TEMPERATURE (T_{stg})	-65 to +150°C
LEAD TEMPERATURE (DURING SOLDERING, FOR 10 s MAX.)	
At distance 1/16 ± 1/32 in. (1.59 ± 0.79 mm) from case for 10 s max	+265°C
Unit case, equipped with a PC Board (min. thickness 1/16 in., 1.59 mm), with solder contacting lead tips only	+300°C

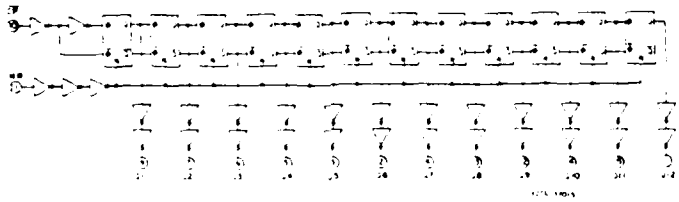


Fig. 1 - Logic block diagram.

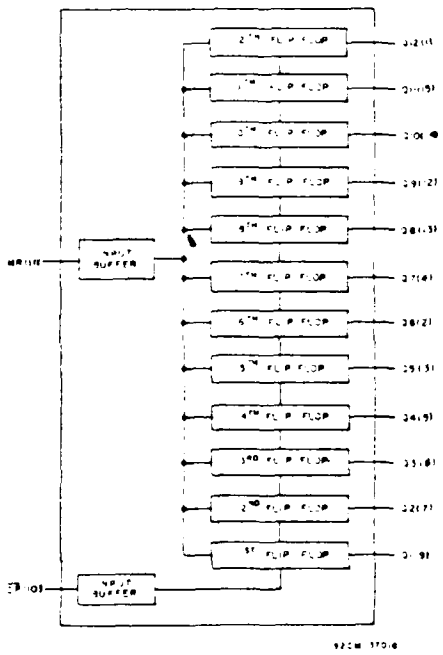


Fig. 2 - Function diagram.

TRUTH TABLE

$\overline{CP}$	MR	Output State
↓	L	No Change
↑	L	Advance to next state
X	H	All Outputs are low

H = high level (steady state)
L = low level (steady state)
X = don't care

RECOMMENDED OPERATING CONDITIONS:
For maximum reliability, nominal operating conditions should be selected so that operation is always within the following ranges:

CHARACTERISTIC	LIMITS MIN.	LIMITS MAX.	UNITS
Supply-Voltage Range (For T_A=Full Package Temperature Range) V_{CC} *			
CD54/74HC Types	2	6	V
CD54 74HCT Types	4 5	5 5	V
DC Input or Output Voltage V_{in}, V_{out}	0	V_{CC}	V
Operating Temperature T_A			
CD74 Types	-40	+85	°C
CD54 Types	-55	+125	°C
Input Rise and Fall Times t_r, t_f			
at 2 V	0	1000	ns
at 4 5 V	0	500	ns
at 6 V	0	400	ns

*Unless otherwise specified all voltages are referenced to Ground

STATIC ELECTRICAL CHARACTERISTICS

CHARACTERISTIC	CD74HC4040/CD54HC4040 TEST CONDITIONS V_{in} V		V_{CC} V	74HC/54HC Series +25°C Min	Typ	Max	74HC Series -40/+85°C Min	Max	54HC Series -55/+125°C Min	Max	CD74HCT4040/CD54HCT4040 TEST CONS. V_{in} V	V_{CC} V	74HCT/54HCT Series +25°C Min	Typ	Max	74HCT Series -40/+85°C Min	Max	54HCT Series -55/+125°C Min	Max	UNITS
High-Level Input Voltage V_{IH}			2	1 5	—	—	1 5	—	1 5	—	—	4 5	2	—	—	2	—	2	—	V
			4 5	3 15	—	—	3 15	—	3 15	—		5 5								
			6	4 2	—	—	4 2	—	4 2	—										
Low-Level Input Voltage V_{IL}			2	—	—	0 3	—	0 3	—	0 3	—	4 5	—	—	0 8	—	0 8	—	0 8	V
			4 5	—	—	0 9	—	0 9	—	0 9		5 5								
			6	—	—	1 2	—	1 2	—	1 2										
High-Level Output Voltage V_{OH} CMOS Loads	V_{IL} or V_{IH}	I_O=-20μA	2	1 9	—	—	1 9	—	1 9	—	V_{IL} or V_{IH}	4.5	4 4	—	—	4 4	—	4 4	—	V
			4 5	4 4	—	—	4 4	—	4 4	—										
			6	5 9	—	—	5 9	—	5 9	—		—	—	—	—	—	—	—	—	
TTL Loads (Standard Output)	V_{IL} or V_{IH}	I_O (mA) -4	4 5	3 98	—	—	3 84	—	3 7	—	V_{IL} or V_{IH}	4 5	3 98	—	—	3 84	—	3 7	—	V
		-5 2	6	5 48	—	—	5 34	—	5 2	—		—	—	—	—	—	—	—	—	
Low-Level Output Voltage V_{OL} CMOS Loads	V_{IL} or V_{IH}	I_O=20 μA	2	—	—	0 1	—	0 1	—	0 1	V_{IL} or V_{IH}	—	—	—	—	—	—	—	—	V
			4 5	—	—	0 1	—	0 1	—	0 1		4 5	—	—	0 1	—	0 1	—	0 1	
			6	—	—	0 1	—	0 1	—	0 1		—	—	—	—	—	—	—	—	
TTL Loads (Standard Output)	V_{IL} or V_{IH}	I_O (mA) 4	—	—	—	—	—	—	—	—	V_{IL} or V_{IH}	—	—	—	—	—	—	—	—	V
			4 5	—	—	0 26	—	0 33	—	0 4		4 5	—	—	0 26	—	0 33	—	0 4	
		5 2	6	—	—	0 26	—	0 33	—	0 4		—	—	—	—	—	—	—	—	
Input Leakage Current I_{in}	V_{CC} or Gnd		6	—	—	±0 1	—	±1	—	±1	V_{CC} or Gnd	5 5	—	—	±0 1	—	±1	—	±1	μA
Quiescent Device Current I_{CC}	V_{CC} or Gnd	I_{OUT}=0	6	—	—	8	—	80	—	160	V_{CC} or Gnd	5 5	—	—	8	—	80	—	160	μA

SWITCHING CHARACTERISTICS (V_{CC}=5 V, T_A=25°C, Input t_r,t_f=6 ns)

CHARACTERISTIC	SYMBOL	54/74HC/HCT	
		Typical	Units
Propagation Delay $\overline{CP}$ to Q1 (C_L = 15 pF)	t_{PLH} t_{PHL}	15	ns
Propagation Delay Q_n to Q_{n+1} (C_L = 15 pF)	t_{PLH} t_{PHL}	6	ns
Propagation Delay MR to Q1 Output (C_L = 15 pF)	t_{PHL}	18	ns
Input Capacitance	C_{IN}	3.5	pF
Power Dissipation Capacitance*	C_{PD}	50	pF

*C_{PD} is used to determine the power consumption
$PD = C_{PD} V_{CC}^2 f_i + \Sigma C_L V_{CC}^2 f_o M$) where:
$M = 2^1 \ 2^2 \ 2^3 \ \ldots \ 2^{12}$
C_L = output load capacitance
f_i = input frequency

Pre-requisite for Switching Function

CHARACTERISTIC	SYMBOL	V_{CC}	25°C				-40°C to +85°C				-55°C to +125°C				UNITS
			HC		HCT		74HC		74HCT		54HC		54HCT		
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
Maximum Clock Frequency	f_{MAX}	2	6	—	—	—	5	—	—	—	4	—	—	—	MHz
		4.5	30	—	25	—	25	—	20	—	20	—	16	—	
		6	35	—	—	—	29	—	—	—	23	—	—	—	
Clock Pulse Width (Figure 3)	t_w	2	80	—	—	—	100	—	—	—	120	—	—	—	ns
		4.5	16	—	20	—	20	—	25	—	24	—	30	—	
		6	14	—	—	—	17	—	—	—	20	—	—	—	
Reset Removal Time (Figure 4)	t_{REM}	2	50	—	—	—	65	—	—	—	75	—	—	—	ns
		4.5	10	—	10	—	13	—	13	—	15	—	15	—	
		6	9	—	—	—	11	—	—	—	13	—	—	—	
Reset Pulse Width (Figure 4)	t_w	2	80	—	—	—	100	—	—	—	120	—	—	—	ns
		4.5	16	—	20	—	20	—	25	—	24	—	30	—	
		6	14	—	—	—	17	—	—	—	20	—	—	—	

SWITCHING CHARACTERISTICS (C_L = 50 pF, Input t_r,t_f=6 ns)

CHARACTERISTIC	SYMBOL	V_{CC}	25°C				-40°C to +85°C				-55°C to +125°C				UNITS
			HC		HCT		74HC		74HCT		54HC		54HCT		
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
Propagation Delay $\overline{CP}$ to Q1 Output (Figure 3)	t_{PLH} t_{PHL}	2	—	175	—	—	—	220	—	—	—	265	—	—	ns
		4.5	—	35	—	40	—	44	—	50	—	53	—	60	
		6	—	30	—	—	—	37	—	—	—	45	—	—	
Propagation Delay Q_n to Q_{n+1} (Figure 3)	t_{PLH} t_{PHL}	2	—	75	—	—	—	95	—	—	—	100	—	—	ns
		4.5	—	15	—	20	—	19	—	25	—	22	—	30	
		6	—	13	—	—	—	16	—	—	—	19	—	—	
Propagation Delay MR to Q1 Output (Figure 4)	t_{PHL}	2	—	200	—	—	—	250	—	—	—	300	—	—	ns
		4.5	—	40	—	45	—	50	—	56	—	60	—	68	
		6	—	34	—	—	—	43	—	—	—	51	—	—	
Output Transition Time (Figure 3)	t_{PLH} t_{PHL}	2	—	75	—	—	—	95	—	—	—	110	—	—	ns
		4.5	—	15	—	15	—	19	—	19	—	22	—	22	
		6	—	13	—	—	—	16	—	—	—	19	—	—	

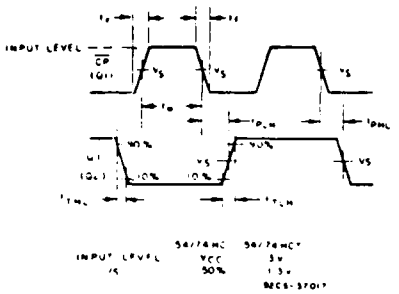


Fig. 3 - Clock pre-requisite times propagation delays and output transition times

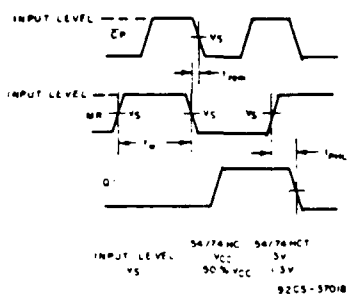


Fig. 4 - Master Reset pre-requisite and propagation delays

CMOS Ripple-Carry Binary Counter/Dividers

High-Voltage Types (20-Volt Rating)

CD4020B – 14 Stage
CD4024B – 7 Stage
CD4040B – 12 Stage

RCA CD4020B, CD4024B, and CD4040B are ripple-carry binary counters. All counter stages are master-slave flip-flops. The state of a counter advances one count on the negative transition of each input pulse; a high level on the RESET line resets the counter to its all zeros state. Schmitt trigger action on the input pulse line permits unlimited rise and fall times. All inputs and outputs are buffered.

The CD4020B and CD4040B types are supplied in 16-lead hermetic dual-in-line ceramic packages (D and F suffixes), 16-lead dual-in-line plastic packages (E suffix), 16-lead ceramic flat packages (K suffix), and in chip form (H suffix).

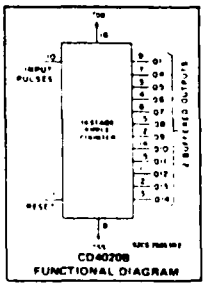
The CD4024B types are supplied in 14-lead hermetic dual-in-line ceramic packages (D and F suffixes), 14-lead dual-in-line plastic packages (E suffix), 14-lead ceramic flat packages (K suffix), and in chip form (H suffix).

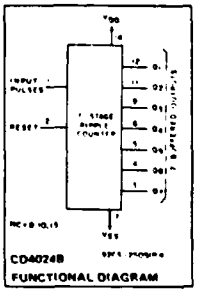
Features:

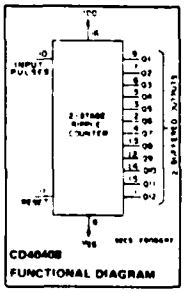
- Medium-speed operation
- Fully static operation
- Buffered inputs and outputs
- 100% tested for quiescent current at 20 V
- Standardized, symmetrical output characteristics
- Fully static operation
- Common reset
- 5-V, 10-V, and 15-V parametric ratings
- Maximum input current of 1 μA at 18 V over full package temperature range; 100 nA at 18 V and 25°C
- Noise margin (over full package temperature range) 1 V at V_{DD} = 5 V; 2 V at V_{DD} = 10 V; 2.5 V at V_{DD} = 15 V
- Meets all requirements of JEDEC Tentative Standard No. 13A, "Standard Specifications for Description of 'B' Series CMOS Devices"

Applications

- Control counters
- Timers
- Frequency dividers
- Time-delay circuits





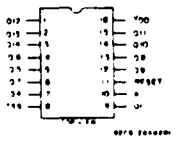


MAXIMUM RATINGS, *Absolute-Maximum Values:*

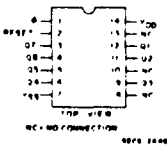
DC SUPPLY VOLTAGE RANGE, (V_{DD}) (Voltages referenced to V_{SS} Terminal)	−0.5 to +20 V
INPUT VOLTAGE RANGE, ALL INPUTS	−0.5 to V_{DD} +0.5 V
DC INPUT CURRENT, ANY ONE INPUT	±10 mA
POWER DISSIPATION PER PACKAGE (P_D)	
For T_A = −40 to +60°C (PACKAGE TYPE E)	500 mW
For T_A = +60 to +85°C (PACKAGE TYPE E)	Derate Linearly at 12 mW/°C to 200 mW
For T_A = −55 to +100°C (PACKAGE TYPES D, F, K)	500 mW
For T_A = +100 to +125°C (PACKAGE TYPES D, F, K)	Derate Linearly at 12 mW/°C to 200 mW
DEVICE DISSIPATION PER OUTPUT TRANSISTOR	
FOR T_A = FULL PACKAGE TEMPERATURE RANGE (All Package Types)	100 mW
OPERATING TEMPERATURE RANGE (T_A)	
PACKAGE TYPES D, F, K, H	−55 to +125°C
PACKAGE TYPE E	−40 to +85°C
STORAGE TEMPERATURE RANGE (T_{stg})	−65 to +150°C
LEAD TEMPERATURE (DURING SOLDERING)	
At distance 1/16 ± 1/32 inch (1.59 ± 0.79 mm) from case for 10 s max	+265°C

TERMINAL ASSIGNMENTS

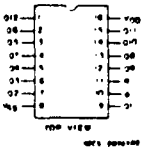
CD4020B



CD4024B



CD40408



RECOMMENDED OPERATING CONDITIONS at T_A = 25°C, Unless Otherwise Specified

For maximum reliability, nominal operating conditions should be selected so that operation is always within the following ranges.

CHARACTERISTIC		V_{DD}	Min.	Max.	UNITS
Supply Voltage Range (at T_A = Full Package Temperature Range)			3	18	V
Input Pulse Frequency,	f_p	5 10 15	– – –	3.5 8 12	MHz
Input Pulse Width,	t_W	5 10 15	140 60 40	– – –	ns
Input Pulse Rise or Fall Time,	t_{rp}, t_{fp}	5 10 15	Unlimited		µs
Reset Pulse Width,	t_W	5 10 15	200 80 60	–	ns
Reset Removal Time,	t_{REM}	5 10 15	350 150 100	– – –	ns

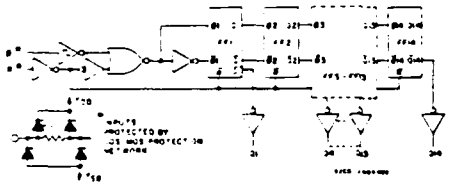


Fig. 1 – Logic diagram for CD4020B.

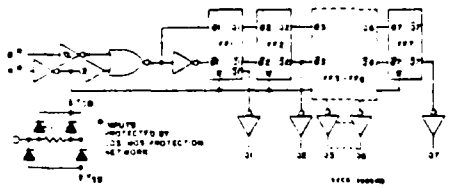


Fig. 2 – Logic diagram for CD4024B.

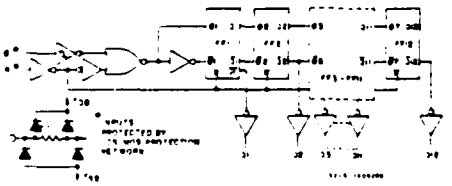


Fig. 3 – Logic diagram for CD4040B.

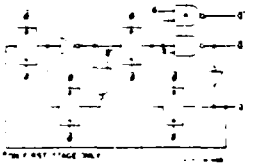


Fig. 4 – Detail of typical flip-flop stage.

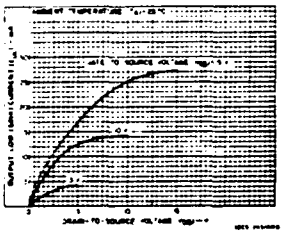


Fig. 5 – Typical output low (sink) current characteristics.

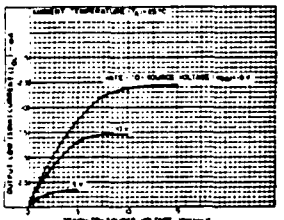


Fig. 6 – Minimum output low (sink) current characteristics.

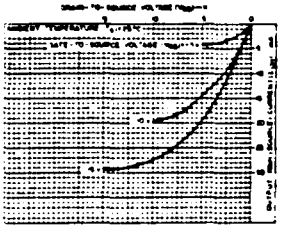


Fig. 7 – Typical output high (source) current characteristics.

CD4020B, CD4024B, CD4040B Types

STATIC ELECTRICAL CHARACTERISTICS

CHARACTERISTIC	CONDITIONS			LIMITS AT INDICATED TEMPERATURES (°C) Values at −55, +25, +125 Apply to D, F, K, H Packages. Values at −40, +25, +85 Apply to E Package							UNITS
	V_O (V)	V_{IN} (V)	V_{DD} (V)	−55	−40	+85	+125	+25 Min	+25 Typ	+25 Max	
Quiescent Device Current I_{DD} Max	–	0,5	5	5	5	150	150	–	0.04	5	µA
	–	0,10	10	10	10	300	300	–	0.04	10	
	–	0,15	15	20	20	600	600		0.04	20	
	–	0,20	20	100	100	3000	3000	–	0.08	100	
Output Low (Sink) Current I_{OL} Min	0.4	0,5	5	0.64	0.61	0.42	0.36	0.51	1	–	mA
	0.5	0,10	10	1.6	1.5	1.1	0.9	1.3	2.6	–	
	1.5	0,15	15	4.2	4	2.8	2.4	3.4	6.8	–	
Output High (Source) Current I_{OH} Min	4.6	0,5	5	−0.64	−0.61	−0.42	−0.36	−0.51	−1	–	
	2.5	0,5	5	−2	−1.8	−1.3	−1.15	−1.6	−3.2	–	
	9.5	0,10	10	−1.6	−1.5	−1.1	−0.9	−1.3	−2.6	–	
	13.5	0,15	15	−4.2	−4	−2.8	−2.4	−3.4	−6.8	–	
Output Voltage Low Level V_{OL} Max		0,5	5	0.05				–	0	0.05	V
	–	0,10	10	0.05				–	0	0.05	
	–	0,15	15	0.05				–	0	0.05	
Output Voltage High Level V_{OH} Min	–	0,5	5	4.95				4.95	5	–	
	–	0,10	10	9.95				9.95	10		
	–	0,15	15	14.95				14.95	15		
Input Low Voltage V_{IL} Max	0.5, 4.5	–	5	1.5				–	–	1.5	V
	1, 9	–	10	3				–	–	3	
	1.5, 13.5	–	15	4				–	–	4	
Input High Voltage V_{IH} Min	0.5, 4.5	–	5	3.5				3.5	–	–	
	1, 9	–	10	7				7	–	–	
	1.5, 13.5	–	15	11				11	–	–	
Input Current I_{IN} Max		0,18	18	±0.1	±0.1	±1	±1	–	$\pm10^{-5}$	±0.1	µA

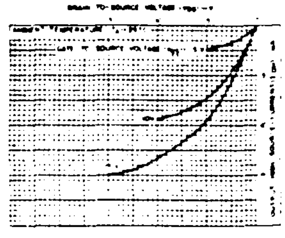


Fig. 8 – Minimum output high (source) current characteristics.

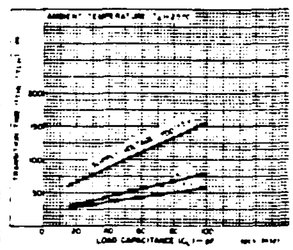
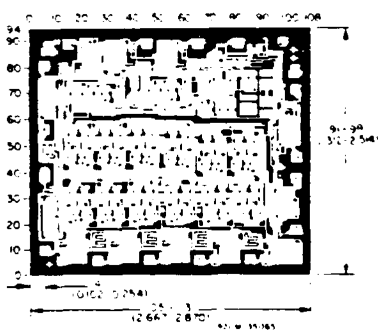
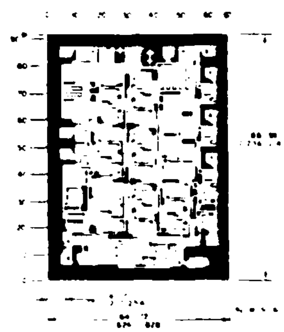


Fig. 9 – Typical transition time as a function of load capacitance.



Dimensions and Pad Layout for CD4020BH. Dimensions and pad layout for CD4040BH are identical.

Dimensions in parentheses are in millimeters and are derived from the basic inch dimensions as indicated. Grid graduations are in mils (10^{-3} inch).



Dimensions and Pad Layout for CD4024BH

The photographs and dimensions of each CMOS chip represent a chip when it is part of the wafer. When the wafer is separated into individual chips, the angle of cleavage may vary with respect to the chip face for different chips. The actual dimensions of the isolated chip, therefore, may differ slightly from the nominal dimensions shown. The user should consider a tolerance of −3 mils to +16 mils applicable to the nominal dimensions shown.

DYNAMIC ELECTRICAL CHARACTERISTICS at T_A = 25°C, Input t_r, t_f = 20 ns, C_L = 50 pF, R_L = 200 kΩ

CHARACTERISTIC	TEST CONDITIONS	V_{DD} (V)	LIMITS Min.	Typ.	Max.	UNITS
Input Pulse Operation						
Propagation Delay Time (Q_1 Out), t_{PHL}, t_{PLH}		5	-	190	360	ns
		10	-	80	160	
		15	-	55	130	
Q_n to Q_{n+1}, t_{PHL}, t_{PLH}		5	-	100	200	ns
		10	-	40	80	
		15	-	30	60	
Transition Time, t_{THL}, t_{TLH}		5	-	100	200	ns
		10	-	50	100	
		15	-	40	80	
Minimum Input Pulse Width, t_W		5	-	70	140	ns
		10	-	30	60	
		15	-	20	40	
Input Pulse Rise or Fall Time, t_{rCL}, t_{fCL}		5	Unlimited			μs
		10				
		15				
Maximum Input Pulse Frequency, f_{CL}		5	3.5	7	-	MHz
		10	8	16	-	
		15	12	24	-	
Input Capacitance, C_I	Any Input		-	5	7.5	pF
Reset Operation						
Propagation Delay Time, t_{PHL}		5	-	140	280	ns
		10	-	60	120	
		15	-	50	100	
Minimum Reset Pulse Width, t_W		5	-	100	200	ns
		10	-	40	80	
		15	-	30	60	
Reset Removal Time, t_{REM}		5	-	175	350	ns
		10	-	75	150	
		15	-	50	100	

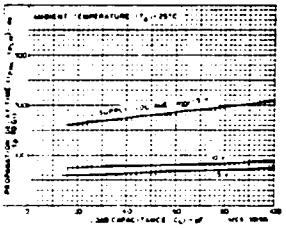


Fig. 10 – Typical propagation delay time as a function of load capacitance (Q to Q_{n+1}).

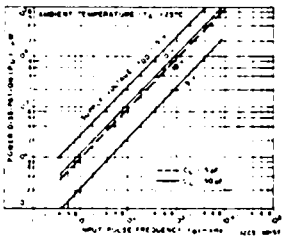


Fig. 11 – Typical dynamic power dissipation as a function of input pulse frequency for CD4020B.

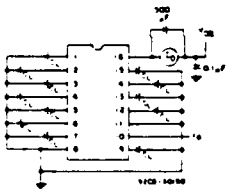


Fig. 12 – Dynamic power dissipation test circuit for CD4020B.

Fig. 13 – Quiescent device current test circuit.

Fig. 14 – Input voltage test circuit.

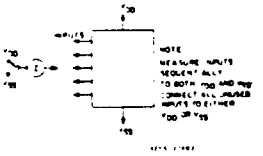


Fig. 15 – Input current test circuit.

Appendix F: Calibration Data

Temperature (C)	Phase Shift Word Hex	Phase Shift Word Dec	T-Chain Count Dec	T-Chain Count Hex	
40.24	E58	3672	1263	4EF	
37.45	E5C	3676	1251	4E4	
36.04	E5B	3675	1247	4DF	
34.60	DD4	3540	1236	4D4	*
30.46	D3D	3389	1221	4C5	*
26.13	DAA	3498	1202	4B2	*
22.23	E30	3632	1182	49E	*
19.43	E98	3736	1166	48E	
16.71	EA2	3746	1153	481	
11.11	EBE	3774	1124	464	
8.30	EBD	3773	1110	456	
5.50	ED3	3795	1095	447	
1.30	EE1	3809	1073	431	
0.08	EE7	3815	1066	42A	
15.83	EA2	3746	1148	47C	
26.29	E03	3587	1202	4B2	*
36.80	E01	3585	1251	4E3	*
43.90	E4D	3661	1277	4FD	
53.14	E4B	3654	1316	524	
55.78	E48	3656	1322	52A	
57.10	E44	3652	1331	533	
57.45	E43	3651	1334	535	
63.14	E3C	3644	1352	548	
64.21	E3B	3643	1359	54F	
64.70	E3D	3645	1361	551	
70.01	E3A	3642	1382	566	

* unstable datapoints marked by asterisks

Vita

William D. Cowan was born on 15 October 1960 in Danbury, Connecticut. He graduated from high school in Lewisburg, West Virginia in 1978. He attended West Virginia Institute of Technology, and graduated in December, 1982. He entered the United States Air Force through Officer Training School and was commissioned in May 1983. He then entered the School of Engineering, Air Force Institute of Technology.

Permanent Address: 11 Melville Ave.
Newtonville, Massachusetts
02160

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approval for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GE/ENG/84D-22	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: School of Engineering, AF Institute of Technology; 6b. OFFICE SYMBOL (If applicable): AFIT/EN	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code): Wright-Patterson AFB OH 45433	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Rome Air Development Center; 8b. OFFICE SYMBOL (If applicable): RADC/EE	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code): Hanscom AFB, Massachusetts 01731	10. SOURCE OF FUNDING NOS: PROGRAM ELEMENT NO. / PROJECT NO / TASK NO / WORK UNIT NO
11. TITLE (Include Security Classification): See Box 19	
12. PERSONAL AUTHOR(S): William D. Cowan, 2Lt, USAF	
13a. TYPE OF REPORT: Thesis; 13b. TIME COVERED: FROM _____ TO _____; 14. DATE OF REPORT (Yr., Mo., Day): 1984, December; 15. PAGE COUNT: 129	
16. SUPPLEMENTARY NOTATION	

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB GR	crystal oscillators, oscillators, surface waves, acoustic waves, frequency stabilizers, compensation, phase shift circuits
9	5		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Approved for public release: IAW AFR 190-17.
LYNN E. WOLAVER 6 Jan 85
Dean for Research and Professional Development
Air Force Institute of Technology (AIC)
Wright-Patterson AFB OH 45433

Title: Development of a Digitally Compensated SAW Oscillator (DCSO) to Provide Temperature Stable Frequency

A single circuit board Digitally Compensated SAW Oscillator (DCSO) running at 500 MHz demonstrated 50 parts per million temperature stability over a portion of its temperature range. A thick film hybrid controller circuit, for use in the DCSO, was designed and tested. This circuit, referred to as the AFIT IC, consists of two commercial TTL prescalers and a custom NMOS integrated circuit. The TTL prescaler circuits of the AFIT IC were found to be operational, but the failure

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS ☐	Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
Roger D. Colvin, Capt, USAF	513-255-3576	AFIT/ENG

DD FORM 1473, 83 APR — EDITION OF 1 JAN 73 IS OBSOLETE.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

(Continued from Block 19)

of the NMOS controller IC to meet design criteria prevented demonstration of a fully integrated DCSO. Off board control circuitry was designed, using portions of the AFIT IC (including part of the NMOS chip), to continue the project. The DCSO was then calibrated and tested over a 0 to 70 degrees Centigrade range. Calibration and testing were facilitated by computer interfaces to the DCSO, an electronic thermometer, and a frequency counter. Despite erroneous temperature behavior of one oscillator loop, compensation to within 30 parts per million over a portion of the temperature range was accomplished for the 500 MHz DCSO.

END

FILMED

5-85

DTIC